APPEARANCE BASED MODELING AND LEARNING OF THE HUMAN FACE WITH APPLICATION TO BIOMETRICS

BY

XUN XU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2010

Urbana, Illinois

Doctoral Committee:

Professor Thomas S. Huang, Chair
Professor Narendra Ahuja
Professor Stephen E. Levinson
Professor Zhi-Pei Liang

ABSTRACT

In this dissertation we study key problems in face processing, with a focus on the applications in biometrics, including both hard biometrics (i.e. conventional face recognition) and soft biometrics, where demographical attributes (e.g. gender and age) are recognized. We categorize face processing techniques into two classes: *appearance-based* approaches that model facial appearance holistically, and *feature-based* approaches, which rely on the localization of facial feature points. In this dissertation we argue that appearance-based approaches are well suited for various face processing tasks.

A fully automatic face processing system consists of several major modules: detection, alignment and recognition. In this dissertation we study the modeling and learning issues in the face alignment and recognition stages, and propose a series of algorithms to tackle these problems, demonstrating how recent developments in machine learning and applied mathematics can be employed to create effective solutions in building fully automatic, appearance-based face processing systems. Although face detection is not covered, as it is a relatively well-solved problem, some algorithms proposed in this dissertation can potentially be applied to that problem as well.

To my family

ACKNOWLEDGMENTS

My Ph.D. study would not have been possible without the support of many people. My deepest gratitude goes to my advisor, Professor Thomas S. Huang, who has been consistently providing guidance, support and help throughout my life at Illinois, both academically and personally. Tom often comments, as modest as he always is, that students are the achievements he is most proud of, but from the viewpoint of us students, working with and learning from him as a role model is truly among the most memorable experiences in our lives. I would like to thank my committee members, Professor Narendra Ahuja, Professor Stephen E. Levinson, and Professor Zhi-Pei Liang, who offered valuable guidance and comments in the preparation of this dissertation. Thanks also go to my collaborators in many research projects, at Illinois or external organizations: Yong Rui, Li-wei He, Dinei Florêncio, Ruei-Sung Lin, Jilin Tu, Ming Liu, Zhenqiu Zhang and Yuxiao Hu. I would also like to express appreciation to my dear fellow IFPers not mentioned above, as well as many other friends, for making this long journey at Champaign-Urbana much more enjoyable than it would be otherwise. Finally, I thank my parents, my wife Shanshan, and my son, Winston — for their endless love, support and tolerance. This dissertation is dedicated to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In this dissertation we study key problems in face processing, with a focus on the applications of biometrics. Here the term biometrics is used in a general sense. We study both hard biometrics that is conventional face recognition, and soft biometrics in which the computer does not attempt to reveal the identity of the person but to analyze certain demographical properties, such as gender, age, and ethnicity group. Hard biometrics has wide applications in access control and surveillance. Soft biometrics, on the other hand, is very useful in scenarios where demographical information is valuable, for example in a retail store where the owner wants to know how female/male customers like certain items.

Generally, a fully automatic face processing system consists of several major components: detection (D), alignment (A) and recognition (R). This framework is sometimes known as the DAR paradigm. The workflow of a DAR face processing system is shown in Figure 1.1.

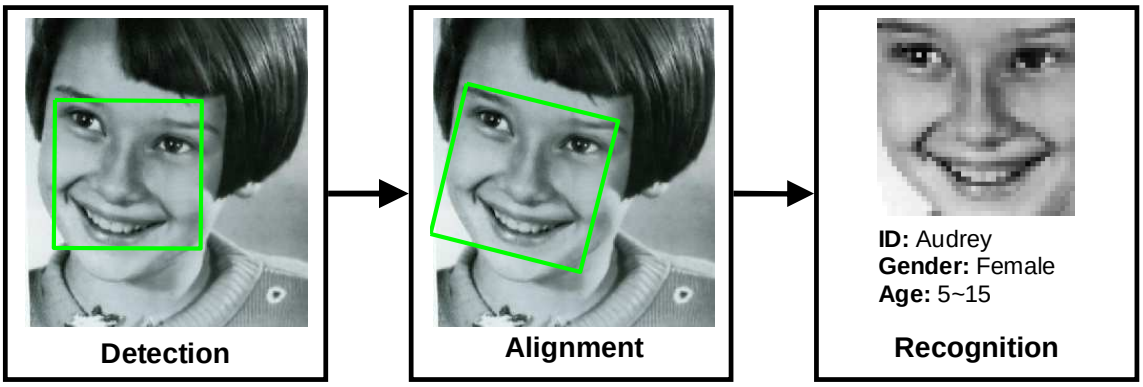


Figure 1.1: Automatic face processing system.

In the face detection stage, the image is analyzed to determine whether faces exist in it, and if so, the location of the face(s) is roughly estimated. The goal of the alignment stage is to obtain a refined estimation of the face's location, so that the cropped face can be processed in the recognition stage, where the actual recognition task is finally conducted.

Face processing techniques can be categorized into two types: *appearance-based* and *feature-based*. We refer to the approaches modeling facial appearance holistically as "appearance-based". These approaches treat the whole face region as a pattern, ignoring the exact location of detailed facial features. On the other hand, "feature-based" approaches rely upon the localization of facial features. The approaches of this category require or involve a step where the facial features, such as corners and contours of the eyes, nose, and mouth, are accurately localized.

Face detection, except for especially difficult cases such as severe occlusion or extreme illumination, is generally regarded as a solved problem. Machine learning–based algorithms, among which the most well-known is Viola-Jones' AdaBoost detector [1], are able to detect faces fairly robustly in most practical scenarios. Besides open source implementations (such as OpenCV [2]) that are freely available, there are stable commercial products as well, such as PittPatt's SDK [3], and many companies have their in-house implementations. Besides the application to biometrics as we study here, face detection is useful in many other applications, such as intelligent surveillance, and in consumer electronic products, such as the face-sensitive auto-focus feature available in many digital cameras available in the market nowadays. Arguably, almost all of the most successful face detection algorithms are appearance-based. Specifically, a common methodology for face detection is to scan a fixed size window across the image where faces need to be detected, and make a face/non-face decision at every location. This converts detection into a classification problem. To this end, these algorithms attempt to

statistically model the difference between the human face and non-face objects by treating the pixels in the fixed size window as a holistic pattern.

In face alignment, most popular approaches are feature-based, which include the Active Shape Model (ASM) [4], Active Appearance Model (AAM) [5], and morphable model [6]. In these approaches, the facial feature points, either sparse or dense, are located in the input image, leading to a detailed representation of both the location and the shape of the face. On the other hand, appearance-based approaches are mainly proposed for the scenario of tracking, such as [7]. However, alignment and tracking are not radically different, both trying to give an accurate estimation of the object's location. Unlike feature-based approaches, appearance-based approaches for alignment do not estimate the exact location of facial features, but they do provide an accurate estimation of the face's overall location, such as translation, in-plane-rotation and scaling.

For face recognition, on the contrary, most existing works are appearance-based, such as Eigenfaces [8, 9], Fisherfaces [10], Bayesian recognition [11] and many others. One typical feature-based face recognition method is the Elastic Bunch Graph Matching (EBGM) algorithm [12], in which facial features are accuratly located and around which wavelet-based descriptors are extracted. The morphable model applied to face recognition [13] can also be categorized as a feature-based approach, although in the recognition stage only the coefficients obtained from alignment are used.

In this dissertation we argue that appearance-based approaches are competent for various face processing tasks. We shall focus on the modeling and learning issues in the face alignment and recognition stages, and propose several machine learning and data-driven algorithms to tackle these problems, demonstrating how recent developments in machine learning and applied mathematics can be employed to create effective solutions in building fully automatic, appearance-based face

processing systems. Face detection is not covered explicitly in this dissertation, as it is a relatively well solved problem. However, some of the the algorithms proposed in this dissertation can potentially be applied to that problem as well.

We first discuss several machine learning algorithms developed for the recognition stage, as they are the core of biometrics systems. In Chapter 2, we introduce the MRC-Boosting algorithm and study how the face recognition (hard biometrics) problem can be tackled efficiently by this algorithm. Then in Chapter 3 we study one soft biometrics problem, gender recognition, presenting the SODA-Boosting algorithm, which is the extension of MRC-Boosting. Chapter 4 discusses the BooMDA algorithm, a novel boosting classification algorithm refining ideas introduced in previous chapters, and potentially applicable to biometrics and other applications. In Chapter 5, we transfer the focus to face alignment, introducing a few appearance-based alignment techniques that were used in our face processing systems. In Chapter 6 we discuss general strategies extending the face processing capability to multiple views, and present empirical study on strategies most relevant to appearance-based face analysis. As a significant module in any practical multi-view face processing system, multi-view alignment is treated in Chapter 7, where we propose an approach based on a tensor appearance model, that is capable of robustly estimating the view and illumination of an input face image, and aligning the face at the same time. In the last chapter, we conclude the dissertation and point out some directions for future research.

CHAPTER 2

FACE RECOGNITION, TARGET DETECTION AND MRC-BOOSTING

We start by discussing the core problem in face-based biometrics: face recognition[1]. Face recognition is usually treated as a multi-class problem. However in this chapter, we are going to show that it can be formulated as a specific class of two-class problem, namely "target detection." To tackle the problem effectively, a novel boosting family classification algorithm called MRC-Boosting is proposed. Through aggregating Maximal Rejection Classifier (MRC) features under the boosting framework, this algorithm can deal with complicated two-class classification problems, especially the category called target detection problems where a target class must be discriminated from the surrounding clutter class. MRC-Boosting is efficient since, in contrast to many other boosting-based algorithms, at each iteration the optimal feature is computed in closed form, with neither an exhaustive search nor a time-consuming numerical optimization. Furthermore, a variant of MRC-Boosting is derived and applied to face recognition. This variant MRC-Boosting algorithm is able to utilize a large number of training samples efficiently, overcoming the difficulty faced by other algorithms such as AdaBoost. The effectiveness of the proposed algorithm is validated by face recognition experiments on several databases, including the publicly available CMU-PIE database [16].

As a problem with theoretical importance and wide applications, automatic face recognition has been actively investigated in the computer vision and pattern

[1]This chapter contains excerpts reprinted from [14] and [15], with permission of IEEE.

recognition community for a long time. Since the well-known Eigenface method [8, 9], many methods have emerged during the past decades, among which an influential one is the Bayesian method proposed by Moghaddam and Pentlend [11]. The essential idea of the Bayesian method is converting face recognition to a two-class discrimination problem for face variation, i.e. classifying the difference between two face images as intrapersonal or extrapersonal. This framework directly models the variation of face images which is most critical for recognition, hence although it is simple and easy to implement, it outperforms many conventional methods. The original Bayesian method assumes Gaussian distribution for both the intrapersonal and extrapersonal differences. However, due to the complicated variation that human faces may have, this assumption may not be valid; thus the Bayesian classifier learned assuming Gaussian distribution for the two classes may not be good enough.

Following such a framework of recognizing faces by discriminating differences, recently more sophisticated classifiers such as SVM [17] and AdaBoost [18, 19] have been employed to solve this problem. AdaBoost is a simple way to build a strong classifier from weak classifiers. However, each iteration of AdaBoost involves searching a large pool of candidate weak classifiers, which is very computationally expensive. On the other hand, this scheme also restricts the best classifier that can be sought (since the chosen classifiers must come from the pool); if the pool does not contain classifiers that are discriminative for the classes under consideration, AdaBoost cannot perform well. Moreover, in the works applying AdaBoost to face recognition [18, 19], another difficulty faced by this algorithm is the huge number of training samples. In both [18] and [19], a resampling strategy is employed to select a small part of samples for training at a time. Although directly utilizing the whole training sample set is preferable, it is infeasible for AdaBoost to do so.

In this chapter, we will show that the face difference discrimination problem

belongs to the category called target detection, where a target class must be separated from the surrounding clutter class. An effective algorithm to tackle problems of this category, namely MRC-Boosting, is proposed, which aggregates Maximal Rejection Classifier (MRC) [20] features in the boosting framework. The merit of this algorithm is that at each iteration, it computes the most discriminative feature in closed form; i.e., there is no exhaustive search (as in AdaBoost) or numerical optimization (as in KL-Boosting [21]) involved. Furthermore, a variant of MRC-Boosting for face recognition is derived, which is able to directly utilize the whole training sample set in an efficient way, thus overcoming the difficulty faced by other algorithms such as AdaBoost.

In Section 2.1, we first briefly review Maximal Rejection Classier and generalize it to the weighted case, then propose MRC-Boosting algorithm. In Section 2.2, face recognition is analyzed as a target detection problem, and a variant of MRC-Boosting that is very efficient for this problem is then derived. Section 2.3 will present experiments on several databases that validate the effectiveness of the proposed method. Following is Section 2.4 where the proposed method is compared to related methods, and Section 2.5 summarizes this chapter.

2.1 MRC-Boosting

2.1.1 Maximal Rejection Classifier (MRC)

Classification is a core problem of pattern recognition, for which linear classifiers are one of the simplest and fastest solutions. The well-known Fisher Linear Discriminant (FLD) assumes that the two classes to be discriminated are linearly separable (or nearly so). For many practical problems, however, this may not be the case. One important category of classification problem is "target detection," in which two classes to be discriminated are called the "target" and the "clutter." In

the feature space, the samples from the target class are surrounded by clutter samples. The goal is to separate (detect) the target from the clutter. Clearly, in this case the two classes are completely not linearly separable. Elad et al. [20] proposed an effective method called Maximal Rejection Classifier (MRC) to solve this problem. MRC is a linear-based classifier that is able to deal with two class problems that are not linearly separable. The formulation of this method is similar to that of FLD; the difference lies in the criterion function. Instead of minimizing the within-class scatter while maximizing between-class scatter as FLD does, MRC tries to find the projection vector that minimizes target scatter while maximizing clutter scatter. Formally, it seeks a vector $\mathbf{w}$ minimizing a functional:

$$\mathbf{w} = \arg\min_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{R}_X \mathbf{w}}{\mathbf{w}^T [\mathbf{R}_X + \mathbf{R}_Y + (\mathbf{m}_X - \mathbf{m}_Y)(\mathbf{m}_X - \mathbf{m}_Y)^T] \mathbf{w}} \tag{2.1}$$

where $\mathbf{m}_X$ and $\mathbf{R}_X$ are the mean and covariance matrix of the target class X respectively, $\mathbf{m}_Y$ and $\mathbf{R}_Y$ are those of clutter class Y. In the case that both X and Y have zero means, (2.1) can be equivalently written as

$$\mathbf{w} = \arg\min_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{R}_X \mathbf{w}}{\mathbf{w}^T \mathbf{R}_Y \mathbf{w}} \tag{2.2}$$

Note that the $\mathbf{R}_X$ in the denominator is also dropped; it is easy to verify that this does not affect the solution. Just as in FLD, the functional to be minimized is a generalized Rayleigh quotient, and the optimal $\mathbf{w}$ can be found through solving a generalized eigenvalue problem and picking the eigenvector associated with the smallest eigenvalue. Intuitively, this formulation makes sense since it tries to find a projection vector on which the target "shrinks" as much as possible, and the clutter is pushed apart as possible, so that the overlapping between the target and the clutter is minimized. Note that in the target detection scenario, it is impossible to separate the two classes on any projection vector (thus FLD cannot work well); the

vector sought by MRC is the optimal one in the sense that it may achieve minimal classification error rate.

After the optimal projection vector $\mathbf{w}$ is found, samples from both classes are projected to $\mathbf{w}$, then *two* threshold values T_1 and T_2 are picked so that as many clutter samples as possible are rejected while all target samples are retained; see Figure 2.1. Note that since two thresholds are used, the decision region for the target using one MRC, i.e. $\{\mathbf{x} : T_1 \leq \mathbf{w}^T\mathbf{x} \leq T_2\}$, is the region between two parallel hyperplanes; therefore MRC is not a linear classifier, but linear-based [20].

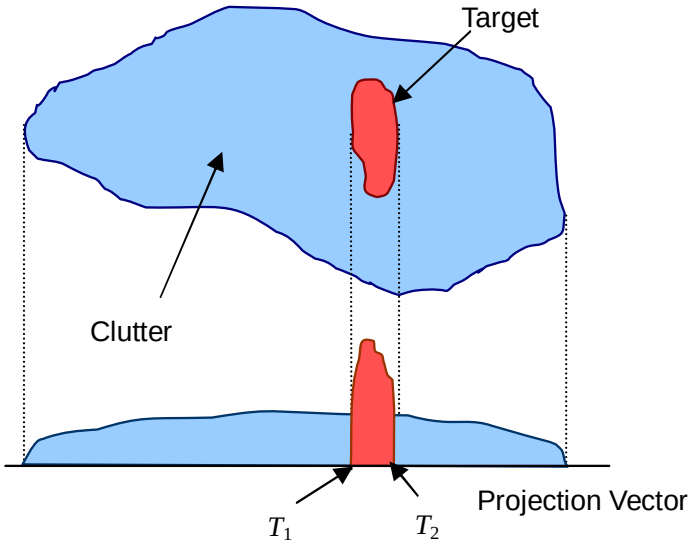


Figure 2.1: Maximal Rejection Classifier.

2.1.2 Weighted MRC

The original MRC assumes that the contribution of each sample is equal, but it can be generalized to the case in which each sample may carry a different weight; thus the sample set represents the underlying distribution of the two classes. Supposing we have a target sample set $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{N_X}\}$ with weights $\{w_1^X, w_2^X, \cdots, w_{N_X}^X\}$ for each $\mathbf{x}_i\,(i = 1, 2, \ldots, N_X)$ respectively, and also a clutter sample set $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{N_Y}\}$ and weights $\{w_1^Y, w_2^Y, \cdots, w_{N_Y}^Y\}$. Since the weights represent a

probability distribution, we have $\sum_{i=1}^{N_X} w_i^X + \sum_{j=1}^{N_Y} w_j^Y = 1$. With the weights, the covariance matrices of two classes are estimated by $\mathbf{R}_X = \sum_{i=1}^{N_X} w_i^X \mathbf{x}_i \mathbf{x}_i^T$ and $\mathbf{R}_Y = \sum_{j=1}^{N_Y} w_j^Y \mathbf{y}_j \mathbf{y}_j^T$. For the simplicity of presentation, we still assume that both classes are of zero-mean. With such definitions, the expression given by (2.2) is directly applicable to finding a weighted MRC.

2.1.3 MRC-Boosting

Boosting is an effective framework for constructing a strong classifier by combining weak component classifiers. The most popular boosting algorithm, AdaBoost, iteratively adds weak classifiers into the ensemble, focusing on the most informative samples, which are not classified well by previously added classifiers. This is done by giving each sample a weight and reweighting each sample at each iteration according to whether it is correctly classified or not at previous iteration. With AdaBoost, the training error of the classifier ensemble can be made arbitrarily low. AdaBoost is a simple and effective way to build a powerful classifier, but its performance and efficiency largely depend on what weak classifiers are employed. In most literature that applies AdaBoost to computer vision problems [1, 18, 19], the "optimal" weak classifiers are found through searching a huge pool (e.g. 52,374 in [18]; 60,480 for [19]) of candidate classifiers, and selecting the one with minimal error rate. This is not only time consuming but also suboptimal, since the candidate pool restricts the classifiers that can be used. Liu and Shum [21] proposed the KL-Boosting algorithm which is able to find an optimal linear feature at each iteration; thus a strong classifier can be built with much fewer weak classifiers compared to AdaBoost. However, taking KL divergence between the two classes' marginal distributions as the objective function, the KL-Boosting algorithm has to employ numerical optimization to find such optimal classifiers which is also computationally expensive. Tu et al. [22] proposed the Fisher-Boosting method employing FLD as

weak classifiers, which can be calculated efficiently. However, FLD cannot work well for highly nonlinearly separable classes such as those in the target detection problem. Therefore, an effective classifier is not likely to be built using Fisher-Boosting for such problems. On the other hand, MRC has the capability of dealing with such non-linearly separable cases. Furthermore, weighted MRCs can be aggregated in a way similar to AdaBoost to construct a strong classifier that is able to tackle complex, target detection-like classification problems. This leads to the MRC-Boosting algorithm, which can be used to discriminate the target (denoted by class label $+1$) from the clutter (denoted by -1), as shown in Algorithm 2.1.

Algorithm 2.1: MRC-Boosting algorithm.

Input: $\left\{(\mathbf{x}_i, c_i), i = 1, 2, \ldots, N : \mathbf{x}_i \in \mathrm{R}^d, c_i \in \{+1, -1\}\right\}$

Initialize: $w_i = \begin{cases} 1/2N^+ & , c_i = +1 \\ 1/2N^- & , c_i = -1 \end{cases}$, where N^+ and N^- are the numbers of the target and clutter samples, respectively. The maximal number of weak classifiers K.

for $k = 1, 2, \ldots, K$ **do**

Find optimal weighted MRC vector $\mathbf{w}$ through solving (2.2), where $\mathbf{R}_X = \sum_{i:c_i=+1} w_i \mathbf{x}_i \mathbf{x}_i^T$ and $\mathbf{R}_Y = \sum_{i:c_i=-1} w_i \mathbf{x}_i \mathbf{x}_i^T$. [2]

Obtain a weak classifier:

$$f_k(\mathbf{x}; \mathbf{w}, T_1, T_2) = \begin{cases} +1 & , T_1 \leq \mathbf{w}^T \mathbf{x} \leq T_2 \\ -1 & , else \end{cases}$$

T_1 and T_2 are determined by minimizing classification error $\varepsilon_k = \sum_{i=1}^{N} w_i I\left(f_k(\mathbf{x}_i) \neq c_i\right)$.

Updating weights:

$$w_i \leftarrow \frac{1}{Z_k} w_i \exp\left[-\alpha_k c_i f_k(\mathbf{x}_i)\right],$$

where $\alpha_k = \frac{1}{2} \ln \frac{1-\varepsilon_k}{\varepsilon_k}$ and Z_k is a normalization factor to ensure $\sum_{i=1}^{N} w_i = 1$.

end

Output: Strong classifier $F(\mathbf{x}) = \mathrm{sgn}\left[G(\mathbf{x})\right]$ where the classification function is $G(\mathbf{x}) = \sum_{k=1}^{K} \alpha_k f_k(\mathbf{x})$.

---

[2] For the simplicity of presentation, here we assume zero-mean for both classes, and the general case contains terms related to the weighted means and is also easy to compute.

Note that the original MRC [20] can construct only a convex (more precisely, a parallelogram polytope) decision region for the target class since the component MRCs are simply combined using an AND rule. In MRC-Boosting, however, the component MRCs are aggregated through weighted voting, so that a much more sophisticated decision bound can be formed.

2.2 Efficient Face Recognition with MRC-Boosting

2.2.1 Face recognition as target detection

The Bayesian method for face recognition [11] is among the most influential ones proposed in the recent years. The essential idea of the Bayesian method is converting face recognition to a two-class discrimination problem for face variation, i.e. classifying the difference between two face images as intrapersonal or extrapersonal. In [11], the two categories are modeled using a Gaussian distribution, and a Bayesian classifier is constructed. With the Bayesian method, face recognition is performed with MAP or Maximum-Likelihood principle, and this can be efficiently implemented as an enhanced eigenface algorithm. Since this method directly models the variation of face images that is most critical for recognition, it outperforms many conventional methods although it is simple and easy to implement. The main drawback of the Bayesian method is that it assumes Gaussian distributions for both the intrapersonal and extrapersonal differences. However, due to the complex variation that people's faces may have under different poses, lighting conditions and expressions, the actual probability distribution of the differences is more complicated than Gaussian. Therefore, the simplified Gaussian assumption restricts the performance of the Bayesian method.

Naturally, one possibility to achieve higher performance is to seek a better

classifier that is more accurate for this complicated two-class discrimination problem. Once such a classifier, say $F(\mathbf{d}) = \text{sgn}[G(\mathbf{d})]$, is sought, we may define the similarity between two faces $\mathbf{F}_1$ and $\mathbf{F}_2$ as the output of the classification function, i.e. $S(\mathbf{F}_1, \mathbf{F}_2) = G(\mathbf{F}_1 - \mathbf{F}_2)$. Larger $S(\mathbf{F}_1, \mathbf{F}_2)$ implies larger positive margin from the decision boundary and that $\mathbf{d} \equiv \mathbf{F}_1 - \mathbf{F}_2$ is more likely to be an intrapersonal difference, therefore $\mathbf{F}_1$ and $\mathbf{F}_2$ are more similar in the sense that they have the same identity. Along this path, more sophisticated classifiers such as the Support Vector Machine (SVM) [17] and AdaBoost [18, 19] have been employed recently. As a simple way to aggregate weak classifiers into a strong classifier, AdaBoost [23] has received much attention in recent years and is applied successfully to face detection [1]. However, as discussed before, it has several drawbacks that restrict its efficiency and effectiveness.

In the two-class discrimination framework for face recognition, we need a good classifier that is able to separate intrapersonal differences from extrapersonal ones. From a geometric point of view, in the image space both intrapersonal and extrapersonal differences have distributions symmetric with respect to the origin because $\mathbf{F}_1 - \mathbf{F}_2$ and $\mathbf{F}_2 - \mathbf{F}_1$ necessarily belong to the same class. Furthermore, intrapersonal differences are surrounded by the extrapersonal ones since most intrapersonal differences have smaller magnitudes than those of extrapersonal ones, as depicted in Figure 2.2. Therefore, this specific two-class problem belongs to the category of target detection where the MRC-Boosting algorithm proposed in subsection 2.1.3 is applicable.

2.2.2 Efficient MRC-Boosting algorithm for face recognition

Although MRC-Boosting given in Algorithm 2.1 seems to be directly applicable, there is another significant problem that should be taken into consideration. Since here the two classes to be discriminated are intrapersonal and extrapersonal

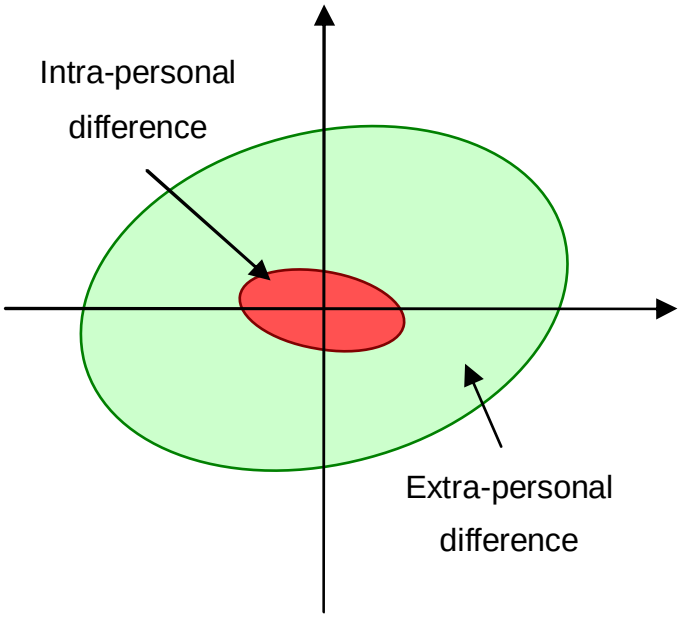


Figure 2.2: Face recognition as target detection: a geometric point of view.

differences, the number of training samples is prohibitively large. For instance, if we have 1000 face images as training data, the total number of the differences between any two of them will be one million. In existing literature employing the AdaBoost algorithm [18, 19], subsampling strategy has to be used to select a small fraction of samples for training at a time, because searching weak classifiers using all the training samples is infeasible. Clearly, utilizing the whole set of training samples is more desirable since it will lead to least biased classifiers. Here we show that with MRC-Boosting, it is possible to directly take into account all the training samples in a computationally efficient way, thus overcoming the problem of a huge training set.

In the case of face recognition, during the training stage we are given a set of training faces $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, with known identities $\{c_1, c_2, \ldots, c_N\}$. That is, the ith and jth faces belong to the same subject if and only if $c_i = c_j$. Taking difference between each pair of training faces generates N^2 differences which constitute the training sample set. As in the general case of MRC-Boosting discussed in subsection 2.1.3, each of these differences, $\mathbf{d}_{ij} \equiv \mathbf{x}_i - \mathbf{x}_j \quad (i, j = 1, 2, \ldots, N)$, carries a weight

w_{ij}. Obviously we should have $w_{ij} = w_{ji}$ since two symmetric differences $\mathbf{d}_{ij}$ and $\mathbf{d}_{ji} = -\mathbf{d}_{ij}$ are equivalent.

In order to find the optimal weighted MRC projection vector at each MRC-Boosting iteration, we should compute the covariance matrix of all weighted intrapersonal differences:

$$\mathbf{S}_I = \sum_{i,j:c_i=c_j} w_{ij} \left(\mathbf{x}_i - \mathbf{x}_j\right) \left(\mathbf{x}_i - \mathbf{x}_j\right)^T$$

and that of the weighted extrapersonal differences:

$$\mathbf{S}_E = \sum_{i,j:c_i\neq c_j} w_{ij} \left(\mathbf{x}_i - \mathbf{x}_j\right) \left(\mathbf{x}_i - \mathbf{x}_j\right)^T$$

Direct computation of these two matrices is very expensive since the computational complexity is $\mathcal{O}(N^2D^2)$ where D is the dimensionality of the face images. Fortunately, they can be computed in a much efficient way. We define D-by-N matrix consisting of the training face vectors $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_N \end{bmatrix}$, N-by-N intrapersonal weighting matrix:

$$\mathbf{W}_I\left(i,j\right) = \begin{cases} w_{ij} & , c_i = c_j \\ 0 & , c_i \neq c_j \end{cases}$$

and extrapersonal weighting matrix:

$$\mathbf{W}_E\left(i,j\right) = \begin{cases} w_{ij} & , c_i \neq c_j \\ 0 & , c_i = c_j \end{cases}$$

It can be shown (see Section 2.6 for the derivation) that:

$$\mathbf{S}_I = 2\mathbf{X}\tilde{\mathbf{W}}_I\mathbf{X}^T \tag{2.3}$$

and

$$\mathbf{S}_E = 2\mathbf{X}\tilde{\mathbf{W}}_E\mathbf{X}^T \tag{2.4}$$

where $\tilde{\mathbf{W}}_I = \mathrm{diag}\,(\mathbf{W}_I\mathbf{e}) - \mathbf{W}_I$ and $\tilde{\mathbf{W}}_E = \mathrm{diag}\,(\mathbf{W}_E\mathbf{e}) - \mathbf{W}_E$. Here $\mathbf{e} = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^T$ and $\mathrm{diag}\,(\mathbf{v})$ denotes a diagonal matrix formed by the elements of vector $\mathbf{v}$. Therefore, with these expressions, $\mathbf{S}_I$ and $\mathbf{S}_E$ can be computed efficiently with complexity $\mathcal{O}(ND^2 + N^2D)$. Compared to direct computation, the saving ratio is $(N + D) : ND$. Considering a typical case $N = D = 1000$, that is $1 : 500$. It can also be noted that usually $\mathbf{W}_I$ is a highly sparse matrix, so that the computation of $\mathbf{S}_I$ is actually less than $\mathcal{O}(ND^2 + N^2D)$.

Through the efficient computation of the intrapersonal and extrapersonal covariance matrices, we may utilize the whole training sample set directly in a computationally feasible way. Finally, this leads to an efficient MRC-Boosting algorithm for face recognition, shown in Algorithm 2.2.

In the recognition phase, the face similarity function $S\,(\mathbf{p}, \mathbf{g})$ defined in Algorithm 2.2 is used to find the most similar gallery face $\mathbf{g}^*$ for a probe $\mathbf{p}$. The classifier $F\,(\mathbf{p}, \mathbf{g}) = \mathrm{sgn}\,[S\,(\mathbf{p}, \mathbf{g})]$ can be used for face verification.

2.3 Experiments

To validate the effectiveness of the proposed MRC-Boosting algorithm for face recognition and to compare it to relevant approaches, we conducted evaluation on several different face databases and scenarios, which are reported in this section.

2.3.1 CMU-PIE database

The first set of experiments were conducted on the CMU-PIE database [16]. This database contains 40,000+ images of 68 subjects; for each person we selected 168

Algorithm 2.2: Efficient MRC-Boosting algorithm for face recognition.

Input: Training faces $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ and known identities $\{c_1, c_2, \ldots, c_N\}$.

Initialize: $w_{ij} = \begin{cases} 1/2N_I & , c_i = c_j \\ 1/2N_E & , c_i \neq c_j \end{cases}$ where N_I and N_E are the numbers of the intrapersonal and extrapersonal differences, respectively. The maximal number of weak classifiers K.

for $k = 1, 2, \ldots, K$ **do**

Compute intrapersonal and extrapersonal covariance matrices $\mathbf{S}_I$ and $\mathbf{S}_E$, via (2.3) and (2.4), respectively.

Find optimal weighted MRC vector $\mathbf{w}$ through solving $\mathbf{w} = \arg\min_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{S}_I \mathbf{w}}{\mathbf{w}^T \mathbf{S}_E \mathbf{w}}$.

Obtain a weak classifier:

$$f_k(\mathbf{x}_i, \mathbf{x}_j; \mathbf{w}, T) = \begin{cases} +1 & , \left|\mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j)\right| \leq T \\ -1 & , else \end{cases}$$

The threshold T is determined by minimizing classification error:

$$\varepsilon_k = \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} I\left(f_k(\mathbf{x}_i, \mathbf{x}_j) \neq \lambda_{ij}\right)$$

where $\lambda_{ij} = \begin{cases} +1 & , c_i = c_j \\ -1 & , c_i \neq c_j \end{cases}$.

Updating weights:

$$w_{ij} \leftarrow \frac{1}{Z_k} w_{ij} \exp\left[-\alpha_k \lambda_{ij} f_k(\mathbf{x}_i, \mathbf{x}_j)\right],$$

where $\alpha_k = \frac{1}{2} \ln \frac{1-\varepsilon_k}{\varepsilon_k}$ and Z_k is a normalization factor to ensure $\sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} = 1$.

end

Output: Strong classifier $F(\mathbf{p}, \mathbf{g}) = \operatorname{sgn}\left[S(\mathbf{p}, \mathbf{g})\right]$, where $S(\mathbf{p}, \mathbf{g}) = \sum_{k=1}^{K} \alpha_k f_k(\mathbf{p}, \mathbf{g})$ is the similarity measure of two faces $\mathbf{p}$ and $\mathbf{g}$.

images that cover large illumination and pose change and moderate variation in expression, constituting a very challenging face database for recognition task. Face images are cropped out from the selected images and resized to be 32×32. Samples from the 11,424 selected face images are depicted in Figure 2.3.

Before our experiments, all these images are normalized to be of zero mean and unit variance. The 168 images of each person are randomly partitioned into 3

Figure 2.3: Samples of CMU-PIE face images used in our experiments. Note that there are large variation in pose, illumination and expression.

disjoint sets: 60 for training, 5 as gallery, and the remaining as probe faces. It should be noted that this experimental setting makes the recognition task fairly challenging due to the small gallery size and large variation in probe faces. An algorithm can perform well in such experiments only if it is capable of learning a model of face variation from the training data, and successfully predicting the possible variation for the limited set of gallery faces. Recognition is performed using MRC-Boosting and other three representative methods: the Bayesian method [11], Eigenface [8, 9], and AdaBoost [18, 19]. First, we compared MRC-Boosting with the Bayesian method to verify whether it can do better in discriminating intrapersonal and extrapersonal differences, and Eigenface is used as a baseline method. MRC-Boosting is trained on the training set; 500 features are obtained in total. The dimensions of both the intrapersonal and extrapersonal subspaces of the Bayesian method are chosen to be 125, a typical value used by [11]. The dimension of Eigenface is also set as 125. The cumulative matching characteristic (CMC) curves [24] of these three methods are shown in Figure 2.4. These curves are obtained by averaging the results from 10 experiments. In each of the experiments, the training images (60 for each person) are fixed, 5 out of the remaining 108 are randomly

chosen as gallery set, and the others are used as probe images. The rank-1 recognition rate of Eigenface only reaches 52.5%, implying the challenge of this experimental setting. The Bayesian method has the rank-1 recognition rate of 80.6%, superior to Eigenface because it directly models face variation which is critical for recognition. Our MRC-Boosting method achieves 86.4%, clearly outperforming the Bayesian method. This result indicates that the MRC-Boosting method is capable of modeling complex face variation better than the Bayesian method does.

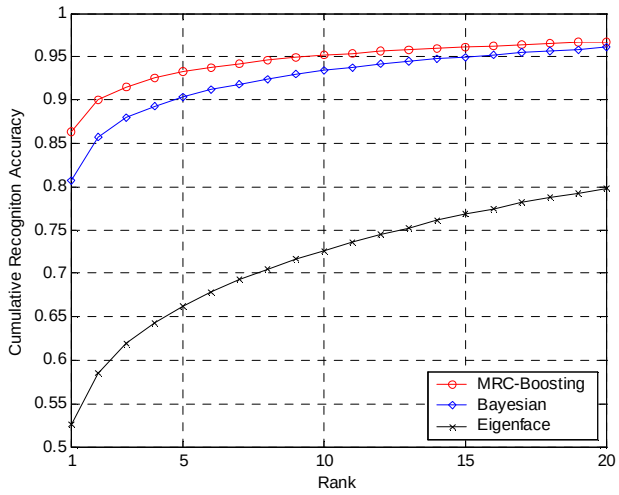


Figure 2.4: The average cumulative matching characteristic (CMC) curves for MRC-Boosting, the Bayesian method, and Eigenface.

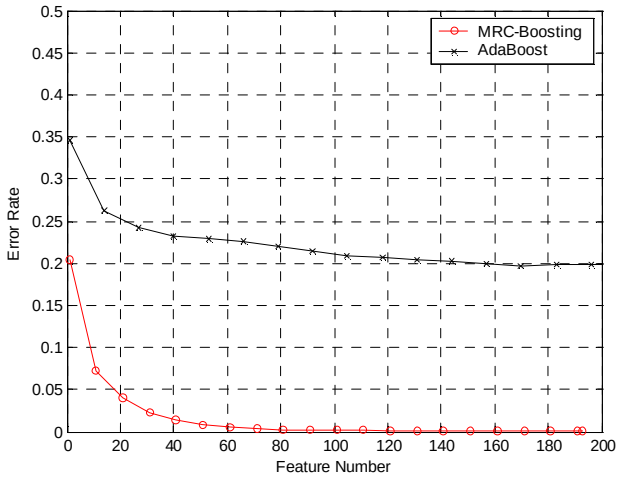
AdaBoost is another algorithm with the potential to build a strong classifier, in order to compare the effectiveness of AdaBoost and MRC-Boosting for face recognition, experiments are performed under the same setting. AdaBoost is trained with the candidate feature pool consisting of both rectangular features and Gabor features, i.e. a superset of those used in [18] and [19]. Since the number of training samples (differences) are prohibitively large (more than 16 million), the resampling strategy suggested by [18] is taken. Figure 2.5(a) shows the curves of both algorithms' error rate on training data with respect to the number of features. As

more and more features are added to the classifier ensemble, a decreasing error rate can be observed for both algorithms. However, the error rate of MRC-Boosting decreases more rapidly than that of AdaBoost, implying that the MRC-Boosting is able to find more discriminative features at each iteration. Figure 2.5(b) shows the curves of the rank-1 recognition accuracy on testing data with respect to the number of features. Again, MRC-Boosting exhibits more rapid increase and ends up with a higher recognition rate than AdaBoost. This indicates that the feature selection mechanism of MRC-Boosting is effective. On the other hand, due to the large variation of faces involved in the experiments, it is difficult for AdaBoost to find most discriminative features from its weak classifier candidates, resulting in its low recognition rate.

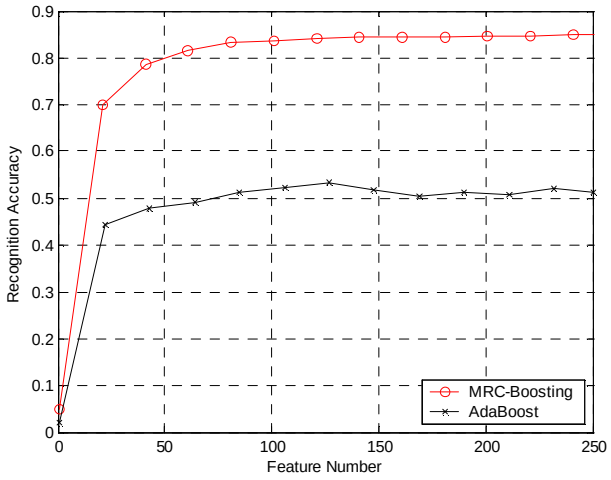
2.3.2 Face recognition in recorded meetings

The second experimentation targets on a practical scenario in which the MRC-Boosting approach was employed to recognize faces in recorded meeting videos. The motivation of this application is to improve user experience in reviewing meeting recordings. Traditionally, off-line meeting reviewing experience is not satisfactory, with most existing systems providing only a linear access to the meeting content. In the past few years, several new meeting recording systems emerged, and the trend is to provide rich and nonlinear access to recorded meetings so that off-line reviewers can have similar experience as those who were in the live meetings. For example, in [25] microphone array sound source localization was used to segment speakers and construct meeting timelines. However, sound source localization can only tell where the sound is coming from; it cannot fulfill tasks like "I want to jump to where my boss John was talking," and this is where face recognition comes in to help.

This work was a collaboration with Microsoft Research, where a meeting



(a) Error rate on the training data vs. the number of features



(b) Face recognition accuracy vs. the number of features

Figure 2.5: Comparison of MRC-Boosting and AdaBoost.

recording device called RingCam was developed to record 360-degree audio and video in a meeting. An example video frame captured by the device is shown in Figure 2.6.

Video sequences (around 9000 frames) captured from three different real meetings using the RingCam were used in the experiments. To evaluate the recognition accuracy, ground truth of the identity and face location of all the meeting attendees was manually labeled every 15 frames. Using the ground truth, the face region

Figure 2.6: Example video frame captured by Microsoft Research RingCam.

images were cropped out from video frames, so that we have a face database containing 14 people, and 3534 images in total, all with the resolution of 24×24. Sample face images are shown in Figure 2.7. It can be observed that there exist large variations in the appearance of the face images, due to partial occlusion (e.g. hands), and the drastic changes of lighting condition, head pose and facial expression (including the effect caused by speaking), all of which are common in real-world meetings. Also noticable is the low resolution of the images of some subjects.

Figure 2.7: Sample images used to evaluate face recognition in recorded meetings.

The 3534 face images were randomly partitioned into three *disjoint* training, gallery and probe sets. 50 images per person (700 in total) were used for training, the gallery contained 10 images for each person, and the rest of the images were used for testing. This experimental setting is quite challenging, since compared to the large variations in the probe images, a gallery size of 10 images/person is rather

small. This setting is intended to simulate the true scenario, where it is often not possible to collect and store a large number of gallery images for each person.

We compared the rank-1 recognition accuracy of MRC-Boosting with Eigenface and the Bayesian method. Before training MRC-Boosting, Principal Component Analysis (PCA) was employed to reduce face images from 576 dimensions to 150. This step was mainly taken to improve robustness, since due to the low resolution and quality (many images appear to be very blurry), the actual dimensionality of the images is much lower than the number of the pixels. For fair comparison, Eigenface also employed a 150-dimensional PCA subspace, and for the Bayesian method, both the intrapersonal and extrapersonal subspaces were of 75 dimensions. The same experiment was performed 20 times, each time with a different random partition of the data. The average and standard deviation of the recognition error rates achieved by three methods are summarized in Table 2.1.

Table 2.1: Error rates of face recognition in recorded meetings.

	Error Rate (%)	Std. Dev. (%)
MRC-Boosting	6.034	0.443
Bayesian	9.592	1.796
Eigenface (PCA)	44.33	1.956

Due to the very large variations in the probe face images, Eigenface did a rather poor job, with an error rate of more than 40%. The Bayesian method was much better, while MRC-Boosting algorithm achieved the best performance. MRC-Boosting also demonstrated higher robustness than the Bayesian method.

Since the recognition is based on a video sequence, instead of many independent still images, the temporal correlation between neighboring frames can be utilized to improve the recognition accuracy. To this end we employed a very simple and fast scheme, which we call identity filtering. It is a postprocessing step after face recognition is done on all the frames independently, where the decision (person identity) on a certain frame results from a voting of decisions on several temporally

neighboring frames. This scheme was applied to all three methods, and the results are shown in Table 2.2.

Table 2.2: Error rates of face recognition in recorded meetings, with identity filtering.

	Error Rate (%)	Std. Dev. (%)
MRC-Boosting	0.724	0.512
Bayesian	1.954	1.838
Eigenface (PCA)	41.35	2.581

The recognition accuracy of all the three methods was improved by this postprocessing scheme. Specifically, the error rate of the MRC-Boosting algorithm was lowered by nearly an order of magnitude. Again, MRC-Boosting was more accurate and robust than both Bayesian and Eigenface methods.

2.3.3 YGA database

In the two previous scenarios, the numbers of persons to be identified are relatively small. Finally we report experimental results on a database named YGA which involves a moderately large number of people. YGA is an internal database used in the University of Illinois at Urbana-Champaign Image Formation and Processing (UIUC-IFP) lab, containing 1600 people with 5 pictures per person. As the pictures were all captured in natural outdoor environments, there exist large variations in illumination and expression.[3] Among the 5 images per person, 4 were randomly selected into the training set, out of which 2 were randomly picked as gallery. Another image was used as probe; therefore the probe set was unseen in the training stage.

We compared our approach to Eigenface and the Bayesian recognition algorithm. The cumulative recognition accuracy curves of the three approaches are shown in Figure 2.8. Again, MRC-Boosting achieved uniformly much higher recognition

[3]Due to the agreement with the data provider, we are not allowed to display the images in publications.

accuracy than both Eigenface and Bayesian approach.

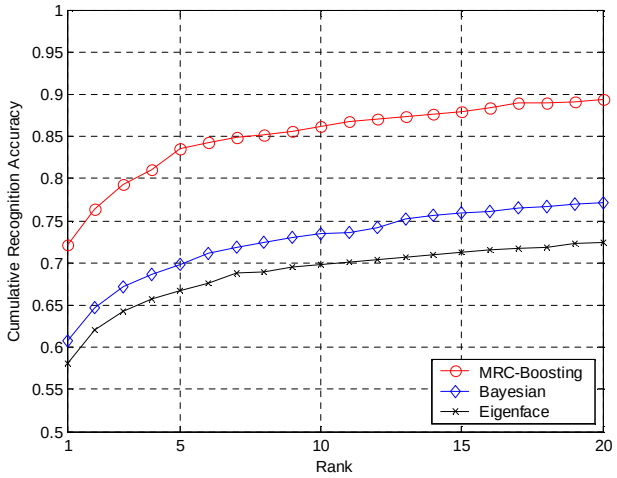


Figure 2.8: Cumulative recognition accuracy on the YGA database.

In Figure 2.9 we show the receiver operating characteristics (ROC) curve for face verification with MRC-Boosting, which gives an equal error rate (EER) of 8.7%; as baseline, Eigenface's curve is also plotted with an EER of more than 20%.

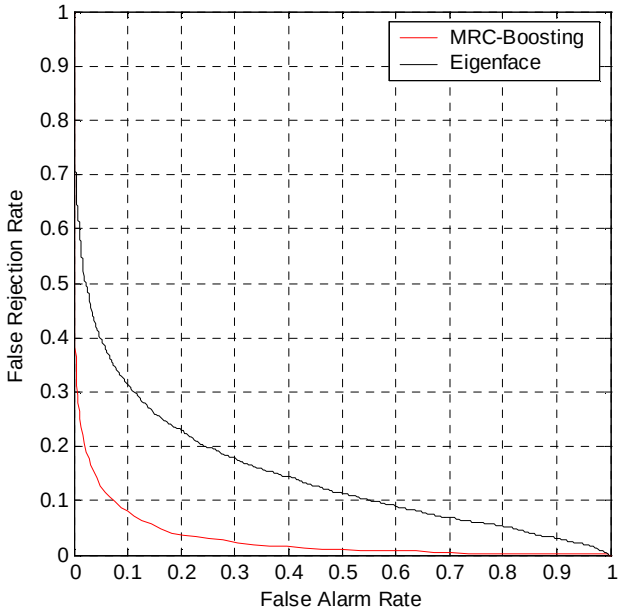


Figure 2.9: ROC curves of face verification on the YGA database.

These results suggest that MRC-Boosting is a promising method for face recognition on relatively large databases. Meanwhile, although not shown by the results, this claim is also supported by MRC-Boosting's efficiency in handling large training sets.

2.4 Discussion

2.4.1 Relationship with LPP

It may be noted that in subsection 2.2.2, the formulation of weighted MRC shares some similarity with Locality-Preserving Projections (LPP) [26]. The optimizations of both methods have similar form, and both can be solved through generalized eigenvalue decomposition. The difference is that MRC seeks a projection vector best discriminating two classes (i.e. intra/extrapersonal differences in the scenario of face recognition), while LPP tries to find a projection vector best preserving the locality relationship between samples in the high dimension space. Also, the similarity matrix S defined in [26] and our intrapersonal weighting matrix $\mathbf{W}_I$ share a similar form, and so do $\tilde{\mathbf{W}}_I$ and LPP's Laplacian matrix L. In LPP the weights in the similarity matrix are set heuristically, whereas in MRC-Boosting the weights are dynamically adjusted through the learning process, automatically focusing on the most informative face pairs.

2.4.2 Appearance vs. 3D model

MRC-Boosting face recognition algorithm is a pure appearance-based method; it models all kinds of variation of human faces in a general framework. 3D model–based methods often demonstrate better recognition performance than appearance-based methods, especially when there exist large variation in pose and

illumination in the face database. For example, in [27] high recognition rates are reported on the CMU-PIE database by fitting a 3D morphable model, better than that of our appearance-based method. However, 3D model–based methods necessarily involve fitting a 3D model to *all* the face images, including every gallery and probe faces, which is an expensive procedure. Moreover, almost all 3D model fitting algorithms need manual initialization. When the size of the face database is large, 3D model–based methods are often unpractical. Therefore, appearance-based methods are usually more efficient.

On the other hand, our MRC-Boosting face recognition algorithm can indeed be combined with 3D model–based methods. We have seen that MRC-Boosting is able to model the variation of faces; actually it can also model the variation of *features*. We may apply MRC-Boosting to analyze the features obtained through 3D model fitting, thus achieving a better classification model than the nearest-neighbor classifier usually used by existing 3D model based methods.

2.5 Summary

In this chapter we proposed MRC-Boosting, an effective classification algorithm aggregating MRC features through the boosting framework. This algorithm is able to handle complicated two-class classification problems, especially the category called target detection problems in which a target class must be discriminated from the surrounding clutter class. The training of MRC-Boosting is very efficient; at each iteration the optimal feature is computed in closed form, which is much more efficient than searching a huge feature pool or numerical optimization as done by many other boosting-based algorithms. Furthermore, we applied MRC-Boosting to face recognition, and proposed a variant of MRC-Boosting that is able to utilize huge amount of training samples efficiently. Face recognition experiments on several

databases, including the CMU-PIE database under a challenging setting, demonstrate the effectiveness of the proposed method.

2.6 Appendix: The Derivation of Equations (2.3) and (2.4)

We consider the general case, in which each difference sample $\mathbf{d}_{ij} \equiv \mathbf{x}_i - \mathbf{x}_j$ is weighted by $\mathbf{W}(i,j) = w_{ij}$. We have:

$$\begin{aligned}
\mathbf{S} &= \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\left(\mathbf{x}_i\mathbf{x}_i^T + \mathbf{x}_j\mathbf{x}_j^T - \mathbf{x}_i\mathbf{x}_j^T - \mathbf{x}_j\mathbf{x}_i^T\right) \\
&= \sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T\sum_{j=1}^{N} w_{ij} + \sum_{j=1}^{N}\mathbf{x}_j\mathbf{x}_j^T\sum_{i=1}^{N} w_{ij} \\
&- \sum_{i=1}^{N}\mathbf{x}_i\sum_{j=1}^{N} w_{ij}\mathbf{x}_j^T - \sum_{j=1}^{N}\mathbf{x}_j\sum_{i=1}^{N} w_{ij}\mathbf{x}_i^T \\
&= 2\left(\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T\sum_{j=1}^{N} w_{ij} - \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\mathbf{x}_i\mathbf{x}_j^T\right) \\
&= 2\mathbf{X}\tilde{\mathbf{W}}\mathbf{X}^T
\end{aligned}$$

where $\tilde{\mathbf{W}} = \mathrm{diag}(\mathbf{We}) - \mathbf{W}$. By substituting $\mathbf{W}$ with $\mathbf{W}_I$ and $\mathbf{W}_E$ as defined in subsection 2.2.2, Equation (2.3) and (2.4) are obtained immediately.

CHAPTER 3

SODA-BOOSTING AND ITS APPLICATION TO GENDER RECOGNITION

In this chapter we extend the MRC-Boosting algorithm discussed in the last chapter, and propose a novel classification algorithm called SODA-Boosting (where SODA stands for Second-Order Discriminant Analysis)[1]. SODA-Boosting is a generic classification problem; as an application, in this chapter it is employed for image-based gender recognition. Experimental results on the publicly available FERET database are reported. The proposed algorithm achieved accuracy comparable to state-of-the-art approaches, and demonstrated superior performance to relevant boosting-based algorithms.

Soft biometrics (automatic recognition of demographic properties, e.g. gender, age and ethnicity, rather than individual identity) from face images has many applications in intelligent surveillance, demographic statistics and human-computer interaction. Since early 1990s, gender recognition has attracted considerable attention from the computer vision and pattern recognition community for a long time. Early works on this topic were mostly based on neural networks [29, 30, 31, 32], where promising performances (more than 90% in accuracy) were reported, although most experiments were conducted on rather small databases (consisting of dozens of images, except [32] where the FERET database was used). From the aspect of pattern recognition, gender recognition is a typical two-class problem. In recent years, the two most successful "off-the-shelf" classifiers, SVM [33, 34] and AdaBoost [35, 36, 37], seem to have dominated in this area, because of

[1]This chapter contains excerpts reprinted from [28], with kind permission of Springer Science+Business Media.

their higher accuracy and robustness compared to earlier techniques. Both classifiers achieve comparably good recognition accuracy [37]. However, AdaBoost-based gender recognizers are generally faster than those based on SVM, which may be a desirable advantage for real-time applications.

This SODA-Boosting algorithm discussed in this chapter is along the AdaBoost line. The main contribution lies in the methodology to discover the most discriminative features in an effective and efficient way. AdaBoost [23] is among the most influential recent advances in machine learning and has been widely adopted in computer vision problems. AdaBoost provides an elegant framework to aggregate weak classifiers into a strong one with theoretically provable generalization performance. In the computer vision community, the most common practice of applying AdaBoost is defining a huge pool of candidate weak classifiers and, in each iteration, seeking the best one through exhaustively traversing the whole feature pool. In order to evade the drawbacks of that approach such as computational load, the proposed algorithm takes a different approach, attempting to directly compute the discriminative features. The proposed algorithm is related to recent algorithms [22, 14] that are based on similar motivation. However, SODA-Boosting makes more comprehensive hypotheses on the distribution of the two classes, resulting a stronger learning procedure. The effectiveness of the proposed algorithm is demonstrated by gender recognition experiments reported in this chapter, where it achieved accuracy comparable to state-of-the-art gender recognizers, surpassing related boosting algorithms in performance.

The rest of this chapter is organized as follows: In Section 3.1 the SODA-Boosting algorithm is introduced and discussed in detail. Section 3.2 presents experimental results of gender recognition on the FERET database [24], where SODA-Boosting is compared to many other algorithms. Finally we summarize the chapter in Section 3.3.

3.1 SODA-Boosting

Boosting, especially AdaBoost (*Ada*ptive *Boost*ing) [23], is one of the most important and influential recent advances in machine learning, regarded by some researchers as the "best off-the-shelf classifier." During recent years, it has been widely adopted in computer vision, resulting in many successful applications. As a meta-algorithm that constructs a strong classifier by "boosting" weak classifiers, the key to employing AdaBoost is to design appropriate "weak classifiers" (or "weak hypotheses"). In the computer vision community, the most common practice is defining a huge pool of candidate weak classifiers (depending on the domain knowledge for the problem of interest), and in each iteration seeking the best one (leading to the lowest classification error rate on the weighted training sample set) through traversing the whole feature pool. Such a practice actually treats AdaBoost as a feature selector. This approach has been popularized since successful deployment of the Viola-Jones face detector [38]. However, as mentioned in Chapter 2, it has several drawbacks:

- The predefined feature pool restricts the optimality of the features that can ever be discovered, as the features that will be incorporated into the final classifier are strictly limited to this pool. For example, if linear weak classifiers are the candidates (which is the case for most applications of AdaBoost in computer vision), the feature pool is indeed only a very sparse sampling of the image space (due to the high dimensionality of the space). Even when the feature pool is "over-complete" (implying that its size is larger than the dimensionality of the data space), there are still too few features to cover the whole space.

- It is computationally expensive. Usually the feature pool is huge, in order to avoid over-sparse sampling of the feature space. Exhaustiveky traversing such

a huge pool in every iteration is clearly time-consuming.

- The design of candidate weak features largely relies on certain domain knowledge of the problem at hand. Therefore the resulting classifier is not generic; i.e. it could not be applied to other problems of different nature. For instance, the Viola-Jones classifier [38], where rectangular filters are used as weak classifiers, cannot be used in an audio related classification problem where the samples to be classified are Mel-frequency cepstral coefficients (MFCCs).

Unlike the conventional practice discussed so far, SODA-Boosting takes another path to discover the weak classifiers. Instead of predefining a feature pool and *searching* for "good" features, it attempts to directly *compute* the weak classifiers that are ideal for the classification purpose, in a computationally efficient way. In the SODA-Boosting algorithm, we limit the weak classifiers to be "linear-based"; i.e., each weak hypothesis is reached by linearly projecting the sample onto a certain vector and thresholding the projection. The key idea in SODA-Boosting is how to learn such linear projection vectors and their corresponding thresholds, which is detailed in the following subsection.

3.1.1 SODA: Second-Order Discriminant Analysis

In the training procedure of AdaBoost, in each boosting iteration one needs to learn a weak classifier to classify the weighted training samples. In our case, the goal is to seek a linear projection and construct an effective weak classifier on that projection. Clearly, the weak classifier should have sufficient discriminative power. In SODA-Boosting, we seek such discriminative linear projections via two different techniques, the Fisher Linear Discrimiant (FLD) and Maximal Rejection Classifier (MRC). With both techniques, the optimal linear projections can be computed in

closed form without exhaustive search or ad hoc numerical optimization. As we shall see, since both FLD and MRC seek discriminative linear projections by utilizing statistical moments of (up to) second-order, we categorize them under a common name SODA (Second-Order Discriminant Analysis).

Fisher Linear Discriminant (FLD)

FLD is the most well-known technique to find a discriminative linear projection [39]. Suppose we need to classify two classes $\mathbf{X}^+, \mathbf{X}^- \subset \mathcal{R}^n$. In the training stage we have a labeled sample set $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$ where $y_i \in \{+1, -1\}$ indicating $\mathbf{x}_i \in \mathbf{X}^+, \mathbf{X}^-$ respectively. In the boosting framework, each training sample is associated with a weight, say $\{w_1, w_2, \ldots, w_N\}$. In each iteration, the weights are normalized so that $\sum_{i=1}^{N} w_i = 1$. The weighted means of the two classes are given by

$$\mathbf{m}_+ = \frac{1}{\sum_{y_i=+1} w_i} \sum\nolimits_{y_i=+1} w_i \mathbf{x}_i \tag{3.1}$$

and

$$\mathbf{m}_- = \frac{1}{\sum_{y_i=-1} w_i} \sum\nolimits_{y_i=-1} w_i \mathbf{x}_i, \tag{3.2}$$

respectively. The weighted scatter matrices are

$$\mathbf{S}_+ = \sum\nolimits_{y_i=+1} w_i (\mathbf{x}_i - \mathbf{m}_+)(\mathbf{x}_i - \mathbf{m}_+)^T \tag{3.3}$$

and

$$\mathbf{S}_- = \sum\nolimits_{y_i=-1} w_i (\mathbf{x}_i - \mathbf{m}_-)(\mathbf{x}_i - \mathbf{m}_-)^T. \tag{3.4}$$

Defining the within-class scatter matrix

$$\mathbf{S}_W = \mathbf{S}_+ + \mathbf{S}_- \tag{3.5}$$

and the between-class scatter matrix

$$\mathbf{S}_B = (\mathbf{m}_+ - \mathbf{m}_-)(\mathbf{m}_+ - \mathbf{m}_-)^T, \tag{3.6}$$

FLD is the vector $\mathbf{w}_{FLD}$ that minimizes criterion

$$J_{FLD}(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}. \tag{3.7}$$

And it turns out that

$$\mathbf{w}_{FLD} = \mathbf{S}_W^{-1}(\mathbf{m}_+ - \mathbf{m}_-). \tag{3.8}$$

The weak classifier associated with the FLD feature is given as

$$f_{FLD}(\mathbf{x}; \mathbf{w}_{FLD}, T) = \begin{cases} +1 & , \mathbf{w}_{FLD}^T \mathbf{x} > T \\ -1 & , else \end{cases}, \tag{3.9}$$

where optimal threshold T is chosen to minimize the classification error

$$\varepsilon = \sum\nolimits_{f_{FLD}(\mathbf{x}_i) \neq y_i} w_i \tag{3.10}$$

Maximal Rejection Classifier (MRC)

The FLD works well when the two classes are linearly separable (or at least approximately so) and when their covariance matrices are close to each other. In many practical problems, these conditions are not met. One especially interesting case is the *target detection* configuration [20], mentioned in Chapter 2, in which one class (the *target*) is surrounded by the other (the *clutter*), as illustrated in

Figure 2.1. This configuration is common in many computer vision problems, among which object detection is a natural example. As the two classes are by no means linearly separable, the best a linear projection can do to discriminate the two classes is to minimize the overlap between the projected samples from the two classes. An intuitive way to achieve this is seeking a projection vector on which the target class is "squeezed," while the clutter class is "pushed aside" as much as possible. Based on this idea, Elad et al. [20] proposed a technique called Maximal Rejection Classifier (MRC) to find the discriminative projections, and applied it to face detection. In Chapter 2 we showed that face recognition can also be modeled as a two-class problem of this type and proposed the MRC-Boosting algorithm where the MRC classifiers are aggregated via boosting.

If we treat class $\mathbf{X}^+$ as the target and $\mathbf{X}^-$ as the clutter, the MRC feature is the projection vector $\mathbf{w}_{MRC+}$, which minimizes the criterion functional

$$J_{MRC+}(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_+ \mathbf{w}}{\mathbf{w}^T (\mathbf{S}_W + \mathbf{S}_B) \mathbf{w}}. \tag{3.11}$$

This criterion seems much similar to that of FLD (3.7), as it is also in the form of a generalized Rayleigh quotient. However, these two classifiers aim at completely different goals, leading to rather different projection vectors. The MRC feature $\mathbf{w}_{MRC+}$ is found through solving a generalized eigenvalue problem and picking the generalized eigenvector associated with the smallest eigenvalue. The weak classifier associated with MRC feature $\mathbf{w}_{MRC+}$ is

$$f_{MRC+}\left(\mathbf{x}; \mathbf{w}_{MRC+}, T_1^+, T_2^+\right) = \begin{cases} +1 & , T_1^+ \leq \mathbf{w}_{MRC+}^T \mathbf{x} \leq T_2^+ \\ -1 & , else \end{cases}, \tag{3.12}$$

where the thresholds T_1^+ and T_2^+ are chosen to minimize classification error, similar

to (3.10). Note that unlike the FLD classifier, this weak classifier contains two thresholds, therefore it is not a linear classifier, but "linear-based" [20].

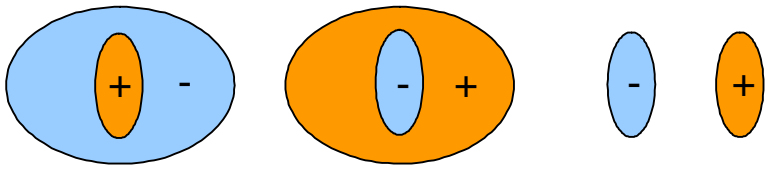
Similarly, if we instead treat class $\mathbf{X}^-$ as the target and $\mathbf{X}^+$ as the clutter, we obtain the other MRC feature $\mathbf{w}_{MRC-}$ which minimizes

$$J_{MRC-}(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_- \mathbf{w}}{\mathbf{w}^T (\mathbf{S}_W + \mathbf{S}_B) \mathbf{w}}, \tag{3.13}$$

and the associated weak classifier

$$f_{MRC-}\left(\mathbf{x}; \mathbf{w}_{MRC-}, T_1^-, T_2^-\right) = \begin{cases} -1 & , T_1^- \leq \mathbf{w}_{MRC-}^T \mathbf{x} \leq T_2^- \\ +1 & , else \end{cases} \tag{3.14}$$

3.1.2 The SODA-Boosting algorithm



(a) MRC+ (b) MRC- (c) FLD

Figure 3.1: Hypotheses made by SODA-Boosting on the distribution of the two classes.

The SODA-Boosting algorithm employs AdaBoost framework to aggregate the SODA classifiers (i.e., FLD and two kinds of MRC classifiers), leading to a strong classifier. In each boosting iteration, because we do not know which SODA feature is appropriate for current distribution of the two classes (represented by the weighted samples), three hypotheses are made, as illustrated in Figure 3.1. The first two hypotheses, MRC+ and MRC−, reflect the cases where $\mathbf{X}^+$ is surrounded by $\mathbf{X}^-$ and vice versa, respectively. The last hypothesis reflects the configuration in which the two classes are well separated and can be reasonably discriminated by FLD. For these hypotheses, we employ techniques discussed in the previous

subsection to obtain corresponding weak classifiers. Naturally, the one resulting in lowest classification error rate best models current distribution of the two classes, hence will be selected and included into the final strong classifier. The algorithm is listed in Algorithm 3.1.

Algorithm 3.1: SODA-Boosting algorithm

Input: $\{(\mathbf{x}_i, y_i), i = 1, 2, \ldots, N : \mathbf{x}_i \in \mathcal{R}^n, y_i \in \{+1, -1\}\}$. The maximal number of weak classifiers K.

Initialize: $w_i = \begin{cases} 1/2N^+ & , y_i = +1 \\ 1/2N^- & , y_i = -1 \end{cases}$, where N^+ and N^- are the numbers of positive and negative samples respectively.

for $k = 1, 2, \ldots, K$ **do**

Compute weighted means $\mathbf{m}_+$, $\mathbf{m}_-$ and scatter matrices $\mathbf{S}_+$, $\mathbf{S}_-$, using (3.1)$\sim$(3.4) respectively.

Compute FLD feature using (3.8), obtain the associated weak classifier $f_{FLD}(\mathbf{x})$ according to (3.9), and calculate its classification error ε_{FLD}.

Compute MRC+ feature by minimizing (3.11), obtain the associated weak classifier $f_{MRC+}(\mathbf{x})$ according to (3.12), and calculate its classification error ε_{MRC+}.

Compute MRC$-$ feature by minimizing (3.13), obtain the associated weak classifier $f_{MRC-}(\mathbf{x})$ according to (3.14), and calculate its classification error ε_{MRC-}.

Select weak classifier $f_k(\mathbf{x}) \in \{f_{FLD}, f_{MRC+}, f_{MRC-}\}$ with minimal classification error ε_k.

Update weights: $w_i \leftarrow \frac{1}{Z_k} w_i \exp\left[-\alpha_k y_i f_k(\mathbf{x}_i)\right]$, where $\alpha_k = \frac{1}{2} \ln \frac{1-\varepsilon_k}{\varepsilon_k}$ and Z_k is a normalization factor to ensure $\sum_{i=1}^{N} w_i = 1$.

end

Output: Strong classifier $F(\mathbf{x}) = sgn\left[G(\mathbf{x})\right]$ where the classification function is $G(\mathbf{x}) = \sum_{k=1}^{K} \alpha_k f_k(\mathbf{x})$.

3.1.3 Discussion

SODA-Boosting is closely related to two other boosting-based classification algorithms: MRC-Boosting [14] that we introduced in Chapter 2, and FisherBoost [22]. MRC-Boosting employs AdaBoost framework to aggregate MRC classifiers. As we demonstrated in the previous chapter, it works well for face recognition. However, like the original MRC approach [20], MRC-Boosting was designed

specifically for *target detection* problems, as illustrated in Figure 2.1. However, for a general two-class discrimination problem, we do not really know whether the distributions of the two classes obey such configuration. Although many problems in computer vision, such as object detection and face recognition, can be modeled as target detection, this assumption is not likely to be true for a general two-class problem. When we do not have a compelling reason to believe that the two classes form a target detection configuration, we would not expect that the features (and the associated weak classifiers) sought by MRC would be effective for classification. On the other hand, FisherBoost does not consider the target detection configuration at all; in each iteration it seeks an FLD classifier to discriminate the two classes. As we know, FLD is effective only when the two classes are linearly separable (or at least approximately so). For a complicated two-class problem, especially after the samples are reweighted as the boosting procedure goes on, the distribution of the two classes may not be such a case. At that point, FLD will fail to discover meaningful discriminants.

SODA-Boosting overcomes the limitation of MRC-Boosting and FisherBoost by including both MRC and FLD classifiers into consideration. FLD and MRC are complementary to each other, working for distinct kinds of configurations of the two classes. Considering all these configurations has much stronger discriminative power than just considering one of them. As we shall see in Section 3.2, putting FLD and MRC classifiers together is actually *not* a trivial combination; it indeed leads to a much stronger learning procedure.

Besides MRC-Boosting and FisherBoost, another relevant approach which also attempts to directly *compute* weak classifiers is KL-Boosting [21]. KL-Boosting discovers discriminative features by seeking linear projection vectors on which the Kullback-Leibler divergence between the two classes is maximized. Unfortunately, such definition of discriminative power cannot lead to a closed form solution; hence

an ad hoc numerical optimization procedure was employed, which is computationally expensive and relatively difficult for implementation. Unlike KL-Boosting, SODA-Boosting (as also MRC-Boosting and FisherBoost) is able to find the discriminative features in closed form, using standard techniques in linear algebra, and hence is more efficient.

3.2 Gender Recognition with SODA-Boosting

We applied the proposed SODA-Boosting algorithm to gender recognition, a typical soft-biometrics application.

The first evaluation was performed on the YGA database (the same database we used in subsection 2.3.3), where we compared the performance of SODA-Boosting to SVM, which is generally considered one of the state-of-the-art approaches to this problem.

In each run of experiments, we randomly partitioned the data into training and test sets. Each time, 80% of the individuals in the database were selected for training; all their images constituted the training set, and the remaining images were used for test. Note that we adopted a protocol similar to that in [37], where training and test data are separated based on individuals, instead of the images themselves, so that no individual in the training set would have any images in the test set. This is in contrast to some earlier work (e.g. [33]) where images of one individual might appear in both training and test set (called "mixed data sets" in [37]), resulting in a more optimistic estimate of recognition accuracy. The protocol employed in our experiments and [37], on the other hand, is closer to the practical scenario where the trained gender recognizer will be tested on images of people it never saw before, leading to more accurate evaluation of the generalized performance.

After 10 random runs, the mean and standard deviation of both approaches' accuracies were calculated as shown in Table 3.1.

Table 3.1: Performance of SODA-Boosting and SVM on YGA database.

Algorithm	**Accuracy**
SODA-Boosting	$85.09 \pm 1.41\%$
SVM	$85.04 \pm 0.93\%$

It needs to be pointed out that the SODA-Boosting model learned from each run's 6400 training images consists of 500 SODA features; whereas the SVM contains more than 3200 support vectors. With a much more compact model, SODA-Boosting achieved performance very comparable to SVM. The smaller model size results in much less computation in the classification stage, which is a significant advantage in real-time applications.

To further compare SODA-Boosting to other related learning algorithms in a detailed way, we conducted another set of experiments on the publicly available FERET database [24]. 4109 frontal face images from the database were aligned according to eye-corner coordinates supplied with the data, and finally normalized to 40×40. In order to normalize the illumination effect, all images were preprocessed to be of zero-mean and unit variance in pixel value. This data set consists of 703 male individuals and 498 female; the ground-truth gender labels were manually labeled by viewing full face images (i.e. before the faces were aligned and cropped out).

Gender recognition experiments were conducted with 10 independent random data partitions; in each run the protocol was the same as before. Average accuracies of different approaches were recorded, as shown in Table 3.2. For comparison purposes, besides the proposed SODA-Boosting algorithm we also reported performance of other classifiers, including SVM, AdaBoost with rectangular filters (used in the Viola-Jones face detector [38]) [35], and two algorithms related to the

proposed one: FisherBoost [22] and MRC-Boosting [14].

Table 3.2: Gender recognition accuracy on the FERET database.

Algorithm	**Accuracy**
SODA-Boosting	92.82%
SVM ($\gamma = 1$)	92.38%
SVM ($\gamma = 10^5$)	93.34%
AdaBoost (Viola-Jones)	92.67%
MRC-Boosting (+)	88.69%
MRC-Boosting (−)	89.76%
FisherBoost	89.54%

For SVM (SVM-Light implementation [40]), a radial basis function (RBF) kernel was used (we also tried polynomial kernels, but the performance was inferior to that of the RBF kernel). As reported by [37], the accuracy is sensitive to parameters C and γ. In our experiments, we took $C = 1/N$ (where $N = 1600$ is the dimensionality of the images), which is a good choice according to results reported in [37], and two different γ values: $\gamma = 1$ which is the default value of SVM-Light and $\gamma = 10^5$ which was shown by [37] to be optimal (through exhaustive parameter tuning). The number of support vectors varies across runs, on average 860 for $\gamma = 10^5$ and 1450 for $\gamma = 1$ respectively. For all the boosting-based algorithms, $K = 500$ weak classifiers were used. The performance of SODA-Boosting (with 500 features) is better than that of SVM with default γ, and is slightly inferior, although still comparable, to SVM with optimal parameter setting (RBF kernel with $\gamma = 10^5$ and $C = 1/1600$). However, note that SVM employs more features (i.e. support vectors) than SODA-Boosting.

Compared to other boosting-based algorithms, SODA-Boosting consistently worked better. For MRC-Boosting, we conducted experiments with two versions (marked with $+/-$) considering male and female as "target" respectively, and both of them achieved inferior performance to SODA-Boosting. In Figure 3.2 we compare the accuracy of different boosting algorithms as the number of weak classifiers

increases. It can be seen that SODA-Boosting clearly exceeded FisherBoost and two versions of MRC-Boosting everywhere. The asymptotic accuracy of SODA-Boosting was very similar to that of the conventional AdaBoost-based algorithms, e.g. [35]. However, thanks to the effort of directly seeking the most discriminative features, SODA-Boosting reached the same accuracy with fewer features than [35]. Although we did not compare SODA-Boosting to the recent approach in [37] directly, we conjecture that the comparison would reach a similar conclusion, as [37] shares the same nature (i.e., boosting very weak features) as [35] and they achieved similar performance. It should be noted that although SODA-Boosting requires fewer features for the same accuracy, it is usually slower than boosting approaches like [35, 37] *in the classification stage*, because those approaches require much less computation per feature. However, those approaches are specifically designed to classify images, relying on weak features that are all predesigned based on certain domain knowledge, hence bounded to certain classification problems. SODA-Boosting, on the other hand, is a generic algorithm, which is, in theory, applicable to any two-class problem, just like SVM.

The point especially worth noticing is that SODA-Boosting clearly exceeded both MRC-Boosting and FisherBoost in performance, although the latter two also aggregate FLD and MRC classifiers, respectively. This convincingly demonstrates that putting together FLD and MRC weak classifiers is *not* a trivial combination; it indeed leads to a stronger learning procedure, as we mentioned in Section 3.1.

3.3 Summary

In this chapter, we proposed a novel boosting-based classification algorithm called SODA-Boosting. Unlike conventional AdaBoost-based algorithms widely used in computer vision (e.g. [38]), SODA-Boosting does not involve any exhaustive search

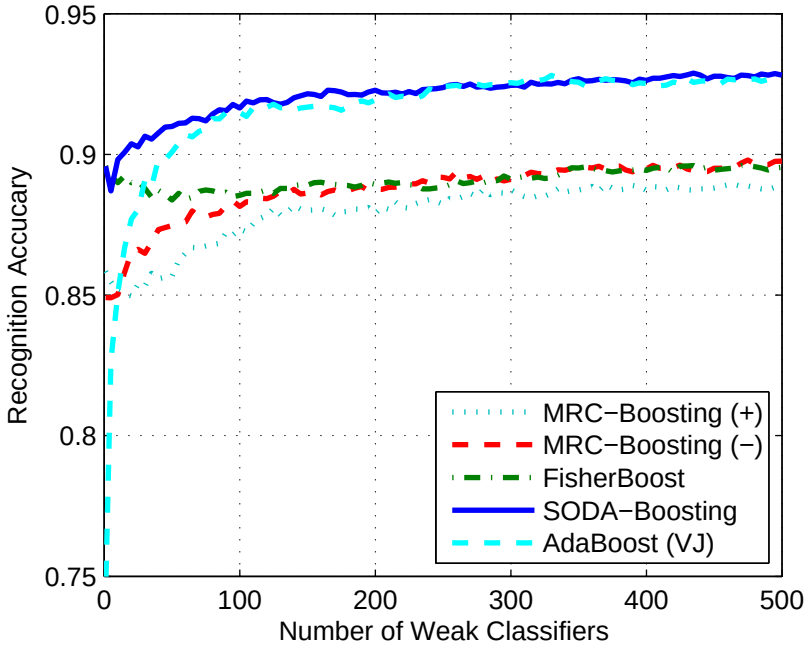


Figure 3.2: Accuracy of different boosting algorithms as the number of classifiers increases.

across a huge feature pool; instead it attempts to directly compute discriminative features in a computationally efficient way. Compared to previous approaches [22, 14] with similar motivation, the proposed algorithm takes into consideration several distinct hypotheses on the distribution of the two classes, resulting a stronger learning procedure. Application to gender recognition shows that SODA-Boosting achieves considerably better performance than those approaches, reaching accuracy comparable to that of state-of-the-art gender recognizers.

CHAPTER 4

BOOMDA: BOOSTED MAXIMAL DIVERGENCE ANALYSIS

In the previous chapters we have shown that discriminative features can be sought by using up to second-order statistics, and these features (namely FLD and MRC) can be aggregated under the boosting framework to obtain fairly powerful classifiers. In every iteration of SODA-Boosting, FLD and MRC features must both be computed and the one achieving the lower error rate is chosen. In this chapter, we first derive a novel algorithm, namely Maximal Divengence Analysis (MDA), to directly seek the discriminative feature, unifying FLD and MRC which are seemingly different approaches. Then we shall also explore how to effectively aggregate MDA features via a boosting procedure, leading to a novel classification algorithm named BooMDA (Boosted MDA). These results refine the ideas gained from the previous chapters.

4.1 Maximal Divergence Analysis

As we have discussed, FLD and MRC are seemingly different algorithms for seeking discriminative linear projections. They work for different cases. FLD is effective when the two classes are linearly separable, or at least approximately so. On the other hand, MRC attempts to find a projection where one class ("target") is centralized, while the other ("clutter") is decentralized. Therefore it is especially effective when the two classes have close means and are not linearly separable, by minimizing the overlap between the two projected classes. The common property of

these two approaches is that both of them utilize up to second-order statistics of the two classes, with the underlying assumption that the two classes obey Gaussian distribution. Now we derive an approach that is also based on the Gaussian assumption but maximizes the separability of the two classes in an explicit and principled way. We shall show that this new approach, which we name MDA (Maximal Divergence Analysis), indeed unifies both MRC and FLD by including them as specific cases.

Assuming that we have two classes obeying Gaussian distributions $\mathcal{N}_1(\mu_1, \Sigma_1)$ and $\mathcal{N}_2(\mu_2, \Sigma_2)$, one way to measure the divergence between these two classes is their Bhattacharyya distance, given as

$$BC(\mathcal{N}_1, \mathcal{N}_2) = \frac{1}{4}\left[(\mu_2 - \mu_1)^T (\Sigma_1 + \Sigma_2)^{-1} (\mu_2 - \mu_1) + \log \frac{|\Sigma_1 + \Sigma_2|^2}{4|\Sigma_1||\Sigma_2|}\right] \tag{4.1}$$

Note that unlike the Kullback-Leibler divergence, the Bhattacharyya distance is symmetric with respect to $\mathcal{N}_1$ and $\mathcal{N}_2$. Our task here is to seek a discriminative linear projection $\mathbf{w}$ upon which the two classes $\mathcal{N}(\mathbf{m}_1, \mathbf{S}_1)$ and $\mathcal{N}(\mathbf{m}_2, \mathbf{S}_2)$ are maximally separated. Employing the Bhattacharyya distance as the index to measure the separability, the most discriminative projection $\mathbf{w}^*$, which we call the MDA (Maximal Divergence Analysis) feature, is given by

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \left(\frac{\left[(\mathbf{m}_2 - \mathbf{m}_1)^T \mathbf{w}\right]^2}{\mathbf{w}^T (\mathbf{S}_1 + \mathbf{S}_2) \mathbf{w}} + \log \frac{\left[\mathbf{w}^T (\mathbf{S}_1 + \mathbf{S}_2) \mathbf{w}\right]^2}{(\mathbf{w}^T \mathbf{S}_1 \mathbf{w})(\mathbf{w}^T \mathbf{S}_2 \mathbf{w})} \right). \tag{4.2}$$

4.1.1 FLD and MRC as specific cases of MDA

It is interesting to note that MDA unifies FLD and MRC in the sense that the latter two are simply included as specific cases of MDA, as we show now.

Case I: $\mathbf{S}_1 = \mathbf{S}_2$

When the two classes share the same covariance matrix, the second term vanishes, and we have

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \frac{\left[(\mathbf{m}_2 - \mathbf{m}_1)^T \mathbf{w}\right]^2}{\mathbf{w}^T (\mathbf{S}_1 + \mathbf{S}_2) \mathbf{w}}$$

If we further assume the two classes have equal prior probability, this is exactly the FLD. It verifies the well-known result that FLD is the optimal (Bayesian) classifier for two Gaussian classes sharing the same covariance matrix.

Case II: $\mathbf{m}_1 \approx \mathbf{m}_2$

When the two classes are closely located (i.e. having similar means), intuitively they are not linearly separable. The first term in (4.2) vanishes, and we have

$$\mathbf{w}^* \approx \arg\max_{\mathbf{w}} \left(\frac{\mathbf{w}^T \mathbf{S}_1 \mathbf{w} + \mathbf{w}^T \mathbf{S}_2 \mathbf{w}}{\sqrt{\mathbf{w}^T \mathbf{S}_1 \mathbf{w}} \sqrt{\mathbf{w}^T \mathbf{S}_2 \mathbf{w}}} \right) = \arg\max_{\mathbf{w}} \left(\frac{\sqrt{\mathbf{w}^T \mathbf{S}_1 \mathbf{w}}}{\sqrt{\mathbf{w}^T \mathbf{S}_2 \mathbf{w}}} + \frac{\sqrt{\mathbf{w}^T \mathbf{S}_2 \mathbf{w}}}{\sqrt{\mathbf{w}^T \mathbf{S}_1 \mathbf{w}}} \right).$$

In order to maximize the objective function, apparently we want either $\mathbf{w}^T \mathbf{S}_2 \mathbf{w} \gg \mathbf{w}^T \mathbf{S}_1 \mathbf{w}$ or $\mathbf{w}^T \mathbf{S}_1 \mathbf{w} \gg \mathbf{w}^T \mathbf{S}_2 \mathbf{w}$ - in other words, "squeezing" one class while "flattening" the other, which is the underlying idea of MRC.

There are differences between MRC and the above specific case of MDA. First, the derivation of MRC explicitly assumes that the *clutter* class has dominating prior probability over the *target* class. However, it is shown to be unnecessary as the derivation of MDA assumes equal prior probabilities of the two classes, but, interestingly, reaches the same result.

On the other hand, to apply MRC, we need to know, a priori, which class is the target and which one is the clutter. This is usually based on some domain knowledge (e.g., for face detection, face is naturally regarded as target and non-face

as clutter), or the knowledge about the distribution of the two classes (which might be obtained through visualization). This is sometimes infeasible because we might know little about the two classes' distribution in the feature space, and usually the visualization of high dimensional data is not practical either. On the contrary, MDA does not need any such knowledge. If the two classes' distribution does form a target detection configuration as defined in [20], the target or clutter will be inferred automatically by the procedure seeking the optimal projection.

4.1.2 Seeking the MDA feature

Letting $\mathbf{S} = \mathbf{S}_1 + \mathbf{S}_2$ and $\mathbf{d} = \mathbf{m}_2 - \mathbf{m}_1$, since $\mathbf{w}$ is defined up to scaling, the optimization problem in (4.2) is equivalent to:

$$\mathbf{w}^* = \underset{\mathbf{w}^T\mathbf{S}\mathbf{w}=1}{\arg\min} \left\{ \log\left[\left(\mathbf{w}^T\mathbf{S}_1\mathbf{w}\right)\left(\mathbf{w}^T\mathbf{S}_2\mathbf{w}\right)\right] - \left(\mathbf{d}^T\mathbf{w}\right)^2 \right\}$$

Applying Lagrange's multiplier approach, we define the Lagrangian function

$$L\left(\mathbf{w},\lambda\right) = \log\left[\left(\mathbf{w}^T\mathbf{S}_1\mathbf{w}\right)\left(\mathbf{w}^T\mathbf{S}_2\mathbf{w}\right)\right] - \left(\mathbf{d}^T\mathbf{w}\right)^2 - \lambda\left(\mathbf{w}^T\mathbf{S}\mathbf{w} - 1\right).$$

Setting its gradient to zero leads us to

$$\left(\frac{\mathbf{S}_1}{\mathbf{w}^T\mathbf{S}_1\mathbf{w}} + \frac{\mathbf{S}_2}{\mathbf{w}^T\mathbf{S}_2\mathbf{w}} - \mathbf{d}\mathbf{d}^T\right)\mathbf{w} = \lambda\mathbf{S}\mathbf{w}.$$

It is easy to see $\lambda = 2 - \left(\mathbf{d}^T\mathbf{w}\right)^2$, therefore the optimal projection vector $\mathbf{w}^*$ is the solution of a non-linear equation

$$\left(\alpha_1\mathbf{S}_1 + \alpha_2\mathbf{S}_2 - \mathbf{d}\mathbf{d}^T\right)\mathbf{w} = \mathbf{0}, \tag{4.3}$$

where $\alpha_i\left(\mathbf{w}\right) = \frac{1}{\mathbf{w}^T\mathbf{S}_i\mathbf{w}} + \left(\mathbf{d}^T\mathbf{w}\right)^2 - 2 \;\; (i = 1, 2)$.

Equation (4.3) does reveal a fact about the MDA projection vector: it lies in the

null space of a matrix that is a linear combination of $\mathbf{S}_1$, $\mathbf{S}_2$, and $\mathbf{d}\mathbf{d}^T$. However, as the combination weights depend on $\mathbf{w}$ in nonlinear way, MDA does not own a closed form solution, as FLD or MRC does. Therefore we have to resort to a numerical method to seek the projection.

We define the Bhattacharyya cost function

$$J(\mathbf{w}) = \frac{1}{2}\left[\log\left(\mathbf{w}^T\mathbf{S}_1\mathbf{w}\right) + \log\left(\mathbf{w}^T\mathbf{S}_2\mathbf{w}\right) - \left(\mathbf{d}^T\mathbf{w}\right)^2\right], \tag{4.4}$$

and the MDA feature is sought by solving

$$\min_{\mathbf{w}^T\mathbf{S}\mathbf{w}=1} J(\mathbf{w}). \tag{4.5}$$

In our implementation, we employ the SQP (Sequential Quadratic Programming) approach [41, Ch.18] to optimize (4.5). The Lagrangian function is

$$\mathcal{L}(\mathbf{w}, \lambda) = J(\mathbf{w}) + \lambda\left(\mathbf{w}^T\mathbf{S}\mathbf{w} - 1\right). \tag{4.6}$$

The gradient of (4.4) can be easily computed as

$$\nabla J = \left(\frac{\mathbf{S}_1}{\mathbf{w}^T\mathbf{S}_1\mathbf{w}} + \frac{\mathbf{S}_2}{\mathbf{w}^T\mathbf{S}_2\mathbf{w}} - \mathbf{d}\mathbf{d}^T\right)\mathbf{w}.$$

Evaluating J and its gradient has $O(N^2)$ complexity, dominated by the computation of $\mathbf{S}_1\mathbf{w}$ and $\mathbf{S}_2\mathbf{w}$.

We only have one simple equality constraint. The constraint function $c(\mathbf{w}) = \frac{1}{2}\left(\mathbf{w}^T\mathbf{S}\mathbf{w} - 1\right)$ can be linearized at current point $\mathbf{w}$ as:

$$c(\mathbf{w}+\mathbf{p}) \approx c(\mathbf{w}) + \nabla c(\mathbf{w})^T\mathbf{p} \tag{4.7}$$

$$= \frac{1}{2}(\mathbf{w}\mathbf{S}\mathbf{w} - 1) + (\mathbf{S}\mathbf{w})^T\mathbf{p} \tag{4.8}$$

Applying the SQP method, at iteration k we need to solve a linear system:

$$\begin{bmatrix} \mathbf{W}_k & \nabla c_k \\ \nabla c_k^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla J_k \\ -c_k \end{bmatrix},$$

where $\mathbf{W}_k = \nabla^2_{\mathbf{ww}} \mathcal{L}(\mathbf{w}_k, \lambda_k)$ is the Hessian of the Lagrangian (4.6). When $\mathbf{W}_k$ is positive definite, this system can be solved as:

$$\lambda_{k+1} = \frac{c_k - \nabla c_k^T \mathbf{W}_k^{-1} \nabla J_k}{\nabla c_k^T \mathbf{W}_k^{-1} \nabla c_k} \tag{4.9}$$

$$\mathbf{p}_{k+1} = -\mathbf{W}_k^{-1} (\nabla J_k + \lambda_{k+1} \nabla c_k) \tag{4.10}$$

For high-dimensional problems that we often encounter in computer vision, evaluating and inverting the Hessian $\mathbf{W}_k$ at each iteration is rather computationally expensive. Instead of working with the exact Hessian matrix, we employ the BFGS quasi-Newton method [41, §8.1], building and updating an approximate inverse Hessian matrix $\mathbf{H}_k \approx \mathbf{W}_k^{-1}$. The BFGS method requires only the gradient of the Lagrangian function, and does not involve matrix inversion — both are highly desirable for our scenario. Substituting $\mathbf{H}_k$ for $\mathbf{W}_k^{-1}$, (4.9) becomes

$$\lambda_{k+1} = \frac{c_k - \nabla c_k^T \mathbf{H}_k \nabla J_k}{\nabla c_k^T \mathbf{H}_k \nabla c_k} \tag{4.11}$$

$$\mathbf{p}_{k+1} = -\mathbf{H}_k (\nabla J_k + \lambda_{k+1} \nabla c_k). \tag{4.12}$$

The BFGS formula for updating $\mathbf{H}_k$ is

$$\mathbf{H}_{k+1} = \left(\mathbf{I} - \rho_k \mathbf{s}_k \mathbf{y}_k^T\right) \mathbf{H}_k \left(\mathbf{I} - \rho_k \mathbf{y}_k \mathbf{s}_k^T\right) + \rho_k \mathbf{s}_k \mathbf{s}_k^T \tag{4.13}$$

$$= \mathbf{H}_k - \rho_k \left(\mathbf{s}_k \mathbf{y}_k^T \mathbf{H}_k + \mathbf{H}_k \mathbf{y}_k \mathbf{s}_k^T\right) + \rho_k \mathbf{s}_k \mathbf{s}_k^T \left(1 + \rho_k \mathbf{y}_k^T \mathbf{H}_k \mathbf{y}_k\right), \tag{4.14}$$

where $\mathbf{s}_k = \mathbf{w}_{k+1} - \mathbf{w}_k = \alpha_{k+1}\mathbf{p}_{k+1}$, $\mathbf{y}_k = \nabla_{\mathbf{w}}\mathcal{L}\left(\mathbf{w}_{k+1}, \lambda_{k+1}\right) - \nabla_{\mathbf{w}}\mathcal{L}\left(\mathbf{w}_k, \lambda_{k+1}\right)$ and $\rho_k = \left(\mathbf{y}_k^T \mathbf{s}_k\right)^{-1}$. The step size α_{k+1} is determined by a suitable line-search procedure, which will be described later. However, this naïve BFGS algorithm requires that $\mathbf{s}_k^T \mathbf{y}_k > 0$ (the curvature condition), which may not hold here. When the curvature condition is not satisfied the BFGS update (4.13) is ineffective. To this end the Damped BFGS Updating algorithm [41, §18.4] is employed. We define

$$\mathbf{q}_k = \mathbf{H}_k^{-1} s_k = -\alpha_{k+1}\left(\nabla J_k + \lambda_{k+1} \nabla c_k\right),$$

then a modified version of $\mathbf{y}_k$ is defined as interpolation between $\mathbf{y}_k$ and $\mathbf{q}_k$:

$$\tilde{\mathbf{y}}_k = \theta_k \mathbf{y}_k + \left(1 - \theta_k\right) \mathbf{q}_k,$$

where

$$\theta_k = \begin{cases} 1 & \text{if } \mathbf{s}_k^T \mathbf{y}_k \geq 0.2 \mathbf{s}_k^T \mathbf{q}_k \\ \frac{0.8 \mathbf{s}_k^T \mathbf{q}_k}{\mathbf{s}_k^T \left(\mathbf{q}_k - \mathbf{y}_k\right)} & \text{if } \mathbf{s}_k^T \mathbf{y}_k < 0.2 \mathbf{s}_k^T \mathbf{q}_k \end{cases}.$$

Notice that by construction $\mathbf{s}_k^T \tilde{\mathbf{y}}_k > 0$. The Damped BFGS update formula is the same as (4.13), simply with $\mathbf{y}_k$ replaced with $\tilde{\mathbf{y}}_k$.

After the search direction $\mathbf{p}_{k+1}$ is computed via (4.11), the step size α_{k+1} should be determined by a line-search procedure. To this end, a merit function must be selected to measure how good a tentative step is in the sense of balancing the decrease of objective function and the violation of the constraint. We choose the ℓ_1

merit function for its simplicity, i.e.

$$\phi(\mathbf{w}) = J(\mathbf{w}) + \mu_{k+1} |c(\mathbf{w})|, \tag{4.15}$$

where μ_{k+1} is the penalty parameter. When $\mu_{k+1} > |\lambda^*|$ where λ^* is the optimal Lagrangian multiplier of (4.6) this merit function is exact; that is, any local solution of (4.5) is its local minimizer [41, §15.3]. Meanwhile, for $\mu_{k+1} > |\lambda_{k+1}|$ (4.15) is guaranteed to be decreasing along the search direction $\mathbf{p}_{k+1}$ obtained via (4.11) [41, §18.5]. In our implementation we select

$$\mu_{k+1} = \max\{\mu_k, 1.1 |\lambda_{k+1}|\} \tag{4.16}$$

with the initial value $\mu_0 = 0$.

For the line-search procedure, the algorithm suggested in [41, §3.4] is chosen, with cubic interpolation. Note that this algorithm requires the directional derivative of (4.15), although the merit function is not differentiable everywhere, it always has a directional derivative [41, §18.5].

4.1.3 Illustrative examples

In this subsection we show a few illustrative "toy" problems to demonstrate that MDA does discover discriminative projections. Specifically, we shall see that MDA is consistently superior to both FLD and MRC. On the other hand, despite its Gaussian assumption, MDA can be fairly effective for non-Gaussian cases as well.

In Figure 4.1, we show examples where discriminative projection vectors are sought to separate Gaussian classes. We visualize the feature vectors obtained via MDA, FLD and MRC (two variants), and show the Bhattacharyya distance of the projected classes in Table 4.1. Apparently, the feature vectors computed via MDA are uniformly more discriminative than those obtained by other approaches.

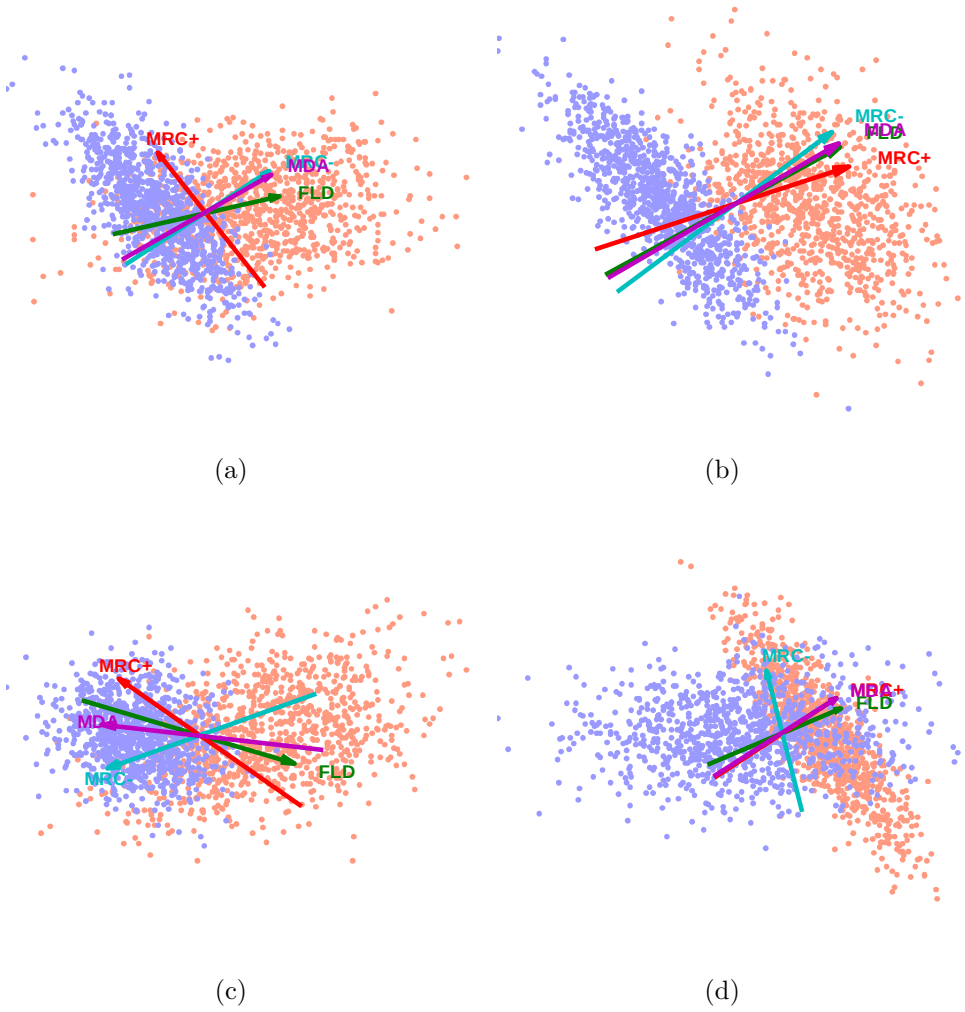


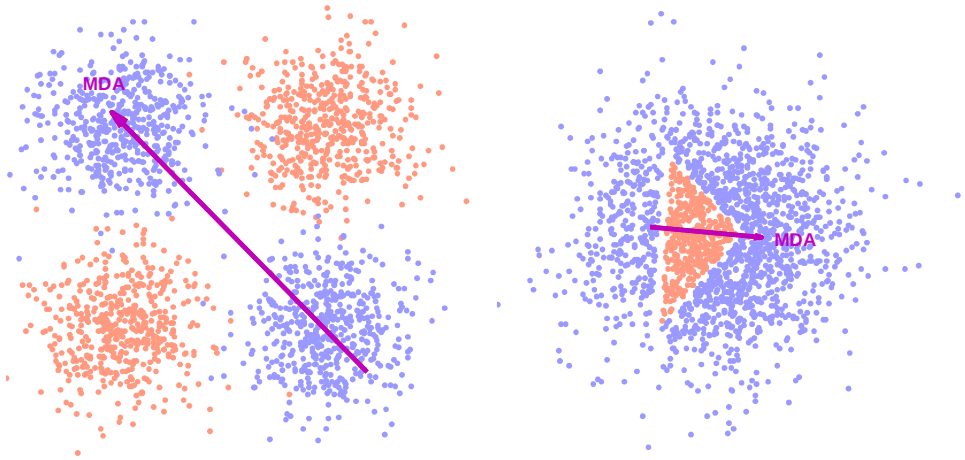
Figure 4.1: Feature vectors sought by MDA, FLD, and MRC to separate Gaussian classes.

Table 4.1: Bhattacharyya distance achieved by feature vectors obtained via MDA, FLD, and MRC.

	FLD	MRC+	MRC-	MDA
(a)	0.4834	0.1654	0.5420	0.5458
(b)	1.5296	1.3754	1.4650	1.5333
(c)	0.6873	0.6066	0.6422	0.6967
(d)	0.4957	0.5534	0.0769	0.5536

Although during the derivation of MDA (as with FLD and MRC) we assume that the two classes obey Gaussian distribution, in practice this assumption need not be strictly valid for this algorithm to work. So long as the means and covariance

matrices supply discriminative information for the two classes, MDA may still be effective in revealing a discriminative projection. In Figure 4.2 we show two non-Gaussian examples and their MDA projection vectors.



(a) XOR (b) Triangle

Figure 4.2: MDA feature vectors for non-Gaussian classes.

4.2 Learning Weak Classifiers

From each MDA feature it is possible to obtain a 1-D classifier, which is clearly not powerful enough for most practical applications. However, such classifiers can be employed as components to build a strong classifier under the boosting framework. There are two key problems here: (1) how to seek MDA features in a boosting setting; (2) how to construct a weak classifier from a MDA feature. The first question has already been answered through the discussion so far. In each boosting iteration, it is straightforward to calculate the means and covariance matrices required by the MDA algorithm, from the weighted set of training samples. For the answer of the second problem, the simplest way is probably thresholding the projection on the MDA feature, reaching a weak classifier with binary output $\{+1, -1\}$. Such weak classifiers can be directly aggregated via the classic discrete

AdaBoost algorithm, just as we have done in the approaches discussed in previous chapters (e.g. MRC-Boosting, SODA-Boosting). However, experimental results [42, 43] have shown that boosting weak hypotheses with confidence-rated output leads to significantly better classification performance.

In [42] Schapire and Singer derived the optimal confidence-rated output for domain-partitioning weak hypotheses (i.e. decision tree). Friedman et al. [43] generalized that result and formulated the Real AdaBoost algorithm, where the weak hypothesis is related to the log-odds on the weighted training set:

$$f_{RAB}(\mathbf{x}) = \frac{1}{2}\log\frac{P_w(y=+1|\mathbf{x})}{P_w(y=-1|\mathbf{x})}, \tag{4.17}$$

which *greedily* minimizes the exponential cost $E_w\left[e^{-yf}\right]$. Note the subscript w that stands for the weights of the training samples, as in each iteration of boosting we are given a training set $\{(\mathbf{x}_i, y_i)\,|i=1,2,\ldots,N\}$ weighted by $\{w_i|i=1,2,\ldots,N\}$.

In [43] Friedman et al. proposed another algorithm, Gentle AdaBoost, suggesting the weak hypothesis in the form:

$$f_{GAB}(\mathbf{x}) = P_w(y=+1|\mathbf{x}) - P_w(y=-1|\mathbf{x}), \tag{4.18}$$

which is numerically more stable as it does not involve either division or logarithm. It was shown in [43] that such weak hypothesis is the Newton step minimizing the exponential cost. Note that the output of f_{GAB} is bounded to $[-1,+1]$, hence "gentler" than f_{RAB}. However, empirically these two algorithms perform similarly well, suggesting that the bounded hypothesis is powerful enough.

In this section we discuss two weak learners: (1) domain-partitioning weak classifier, a commonly used weak classifier, for which we suggest an efficient algorithm to determine the optimal partitioning; and (2) Quasi Logistic Regression weak classifier, a continuous, discriminative classifier minimizing a specific cost

function. These weak classifiers not only output a binary classification decision, but provide the confidence of their decision as well.

4.2.1 Domain-partitioning weak classifier

One commonly used learner is the domain-partitioning weak classifier (i.e. decision tree). In fact, according to [43], AdaBoost with decision trees was referred to as the "best off-the-shelf classifier in the world" by Leo Breiman. The key problem in employing domain-partitioning classifiers is how to determine the optimal domain partitioning scheme. In this subsection we suggest an efficient dynamic programming algorithm for this purpose.

Supposing we split the domain along the projection vector into P partitions; in other words, we build a P-node decision tree based on the projected samples. We call P the *degree* of this domain-partitioning scheme. The partitioning is done by selecting $P-1$ thresholds $\{T_i | i = 1, 2, \ldots, P-1\}$ satisfying

$$-\infty \equiv T_0 < T_1 < T_2 < \cdots < T_{P-1} < T_P \equiv +\infty.$$

Then the weak learner is

$$f(\mathbf{x}) = \sum_{i=1}^{P} c_i \mathbf{1}_{[T_{i-1} \le \mathbf{w}^T \mathbf{x} < T_i]},$$

where c_i is the confidence-rated output for the i-th partition. If we employ the Gentle AdaBoost algorithm, we have

$$c_i = 2P_w\left(y = +1 | T_{i-1} \le \mathbf{w}^T \mathbf{x} < T_i\right) - 1.$$

To determine the optimal domain partitioning, several criteria can be employed, such as cross entropy, Gini index [44, §14.4], and the one suggested in [42] (their

Eqn. 10), while the simplest one is the classification error rate. Since we have only a finite number of choices for the thresholds (any threshold falling between two neighboring projected samples has the same effect), we can search for the optimal partitioning.

Clearly, exhaustive search is computationally prohibitive, even for moderately large P (e.g. 10). Fortunately, all of the criteria mentioned above have the form of summation over partitions, revealing an "optimal substructure." Therefore, a dynamic programming (DP) algorithm can be applied here to determine the optimal domain partitioning efficiently. The DP algorithm for optimal domain partitioning has a complexity of $O\left(N \log N + \max\left\{P-2, 0\right\} N^2\right)$, where N is the number of training samples. If the training set is large, we can build a K-bin histogram so that the Ns in the above expression is replaced with K, reducing the complexity to be linear with respect to N (note that the histogram has to be built in $O\left(N\right)$).

After obtaining the optimal partition, the weak hypothesis suggested by Gentle AdaBoost is

$$f_{GAB}\left(\mathbf{x}\right) = 2\sum_{i=1}^{P} P_w\left(y=+1 | T_{i-1} \leq \mathbf{w}^T\mathbf{x} < T_i\right) \mathbf{1}_{\left[T_{i-1} \leq \mathbf{w}^T\mathbf{x} < T_i\right]} - 1.$$

As we have mentioned, it is a Newton step minimizing the exponential cost $E_w\left[e^{-yf}\right]$. While Gentle AdaBoost is numerically more stable than Real AdaBoost, the latter leads to a *greedy* minimizer of the exponential cost, hence approaching the optimum more quickly. In order to accelerate convergence, we let the weak hypothesis take the form of

$$f\left(\mathbf{x}\right) = \alpha f_{GAB}\left(\mathbf{x}\right) + \beta,$$

where affine parameters α and β are chosen to minimize $E_w\left[e^{-yf}\right]$. This approach can be regarded as a hybrid of Gentle and Real AdaBoost. To determine the optimal

α and β, we write down the gradient and Hessian matrix of the exponential cost:

$$\nabla E_w \left[e^{-y(\alpha f_G + \beta)} \right] = E_w \left[-y e^{-y(\alpha f_G + \beta)} \begin{bmatrix} f_G \\ 1 \end{bmatrix} \right] \tag{4.19}$$

$$\nabla^2 E_w \left[e^{-y(\alpha f_G + \beta)} \right] = E_w \left[e^{-y(\alpha f_G + \beta)} \begin{bmatrix} f_G \\ 1 \end{bmatrix} \begin{bmatrix} f_G & 1 \end{bmatrix} \right] \tag{4.20}$$

Since the Hessian is non-negative definite, this optimization problem is convex, and a simple Newton-Raphson algorithm can be employed to solve it. In practice, we found that the minimum can usually be reached in a few iterations, with initialization $\alpha = 1$ and $\beta = 0$ (usually close to the optimum). We also found that the α value computed as [42, §3.1] suggested often leads to a significantly higher cost, although it does minimize an upper bound of the cost. Therefore, explicit optimization here is a more practical choice.

4.2.2 Quasi Logistic Regression weak classifier

Decision tree (domain-partitioning) weak classifiers are not continuous, resulting in a highly rough strong classifier after they are combined via boosting. For many pattern recognition problems in practice, a continuous classifier is more desirable. For example, in a face recognition system, when the input image is shifted a little (e.g. by subpixel), we hope the output from the classifier would not change drastically. Therefore, it is helpful to design a smoother weak classifier.

As we have mentioned before, the confidence-rated weak classifiers in either Real AdaBoost or Gentle AdaBoost are closely related to the weighted samples' posterior. Supposing the Gaussian assumption for the two classes is valid, then the samples projected upon the learned MDA vector, $z \equiv \mathbf{w}^T \mathbf{x}$, obey Gaussian distribution, say

$\mathcal{N}(\mu_1, \sigma_1)$ and $\mathcal{N}(\mu_2, \sigma_2)$ respectively. The log-odds of the two classes is

$$\log\left[\frac{P_w(y=+1|z)}{P_w(y=-1|z)}\right] = \log\left[\frac{\pi_1 \exp\left(-(z-m_1)^2/2\sigma_1^2\right)}{\pi_2 \exp\left(-(z-m_2)^2/2\sigma_2^2\right)}\right] \tag{4.21}$$

$$= \frac{\sigma_1^2 - \sigma_2^2}{2\sigma_1^2\sigma_2^2} z^2 + \left(\frac{m_1}{\sigma_1^2} - \frac{m_2}{\sigma_2^2}\right) z + \left(\frac{m_2^2}{2\sigma_2^2} - \frac{m_1^2}{2\sigma_1^2} + \log\frac{\pi_1}{\pi_2}\right). \tag{4.22}$$

In other words, the log-odds is a quadratic function of z. We may write it in the form of a generalized linear model:

$$l(z; \mathbf{v}) = \log\left[\frac{P_w(y=+1|z)}{P_w(y=-1|z)}\right] = \phi^T \mathbf{v} \tag{4.23}$$

where the features $\phi = \begin{bmatrix} 1 & z & z^2 \end{bmatrix}^T$. Accordingly, the posterior of sample $\mathbf{x}$ is

$$P_w(y|\mathbf{x}) = P_w(y|z) = \left[1 + e^{-yl(z;\mathbf{v})}\right]^{-1}, \tag{4.24}$$

then the Gentle AdaBoost-style weak hypothesis is simply given by (4.18):

$$f_{GAB}(\mathbf{x}) = 2\left[1 + e^{-l\left(\mathbf{w}^T\mathbf{x};\mathbf{v}\right)}\right]^{-1} - 1. \tag{4.25}$$

This quadratic form makes the classifier able to handle classification problems where one class is surrounded by the other as in the target detection scenario. Notice that assumption (4.23) is indeed weaker than the Gaussian assumption above, as it does not assume the form of the class conditional probability of both classes but only requires the log-odds to be a quadratic function. Indeed, we are not limited to a quadratic function either, but can generalize this representation, letting the log-odds be a polynomial of arbitrary degree; then the features

$$\phi = \left[\begin{array}{cccc} 1 & z & \cdots & z^{P-1} \end{array} \right]^T .$$

The integer P here is also called the *degree* of the weak classifier (4.25). It is analogous to the degree of the domain-partitioning weak classifier, since a degree P classifier in the form of (4.25) is also able to partition the z axis into P regions. A higher-degree representation makes it possible to model highly non-Gaussian classes, resulting in enhanced classification accuracy.

So far we have not covered how to learn the generalized linear coefficients $\mathbf{v}$, neither have we explained why classifier (4.25) is named "Quasi Logistic Regression." Both questions are now to be clarified.

The posterior (4.24) is exactly the one used by logistic regression (LR), so it is intuitive to learn the generalized linear coefficients $\mathbf{v}$ with standard LR technique. Since, in the boosting setting our training samples are weighted, we want to seek $\mathbf{v}$ minimizing

$$J_{LR}(\mathbf{v}) = E_w \left[-\log P_w(y|z) \right] = E_w \left[\log \left(1 + e^{-yl(z;\mathbf{v})} \right) \right] \tag{4.26}$$

whose gradients and Hessian are

$$\nabla J_{LR} = -E_w \left[\frac{y\phi}{1 + e^{yl(z;\mathbf{v})}} \right]$$

and

$$\mathbf{H}_{\mathrm{LR}} = E_w \left[\frac{e^{yl(z;\mathbf{v})}}{\left(1 + e^{yl(z;\mathbf{v})}\right)^2} \phi\phi^T \right]$$

respectively. The Newton-Raphson update is simply $\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} - \mathbf{H}^{-1}\nabla J$. Since $\mathbf{H}$ is always nonnegative definite, the iterations will converge to the global minimum.

Empirically, we found that the weak classifiers learned using logistic regression

did not effectively reduce the training error. An immediate thought is that the discriminant function $l(z;\mathbf{v})$ fitted by LR does not explicitly target classification, due to the cost function it uses (4.26), where the function

$$\log\left(1+e^{-a}\right)$$

does not well approximate the misclassification error cost

$$c(a)=\begin{cases}1 & a\leq 0\\ 0 & a>0\end{cases}.$$

This suggests using a flipped sigmoid

$$s(a)=(1+e^{a})^{-1},$$

or "soft error," as a better approximation, and learn $\mathbf{v}$ by minimizing

$$J_{SE}(\mathbf{v})=E\left[\left(1+e^{yl(z;\mathbf{v})}\right)^{-1}\right]. \tag{4.27}$$

Indeed, there is an underlying justification for employing soft error as the cost function for learning $l(z;\mathbf{v})$. Note that when the Gentle AdaBoost weak hypothesis (4.25) is used in the boosting procedure, the cost we need to minimize at a certain iteration is

$$J_{GAB}=E\left[\exp\left[-y\left(\frac{2}{1+e^{-l}}-1\right)\right]\right] \tag{4.28}$$

$$=E\left[\exp\left(\frac{1-e^{yl}}{1+e^{yl}}\right)\right] \tag{4.29}$$

In Figure 4.3 we plot J_{SE}, J_{LR} and J_{GAB} (scaling is done to make them match with each other at zero). It is clear that J_{SE} is a much better approximation than

J_{LR} to the true cost J_{GAB} that we need to minimize; moreover it has a simpler form than J_{GAB} itself. In other words, the soft error is more consistent with the goal of boosting.

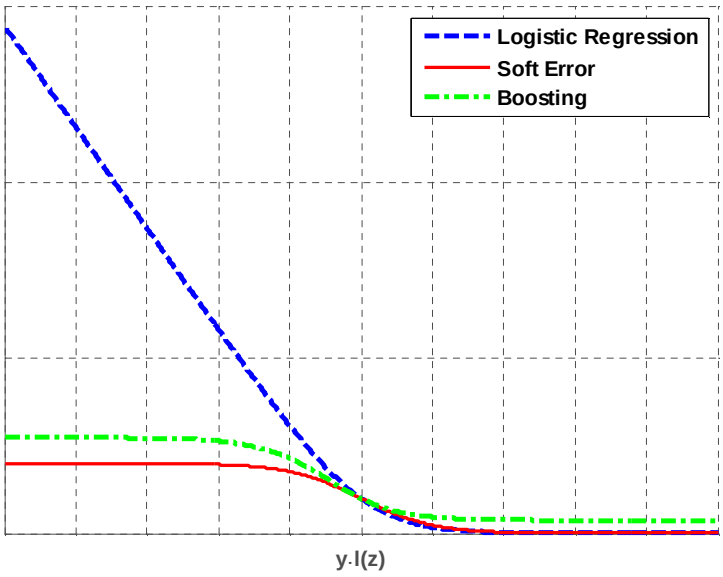


Figure 4.3: Comparison of different cost functions for learning the discriminant function $l(z;\mathbf{v})$.

The gradients and Hessian of (4.27) are respectively

$$\nabla J_{SE} = -E\left[\frac{ye^{yl(z;\mathbf{v})}}{\left(1+e^{yl(z;\mathbf{v})}\right)^2}\phi\right]$$

and

$$\mathbf{H}_{\mathrm{SE}} = -E\left[\frac{\left(1-e^{yl(z;\mathbf{v})}\right)e^{yl(z;\mathbf{v})}}{\left(1+e^{yl(z;\mathbf{v})}\right)^3}\phi\phi^T\right]$$

Because $\mathbf{H}$ is indefinite, this problem does not have a unique global optimum. Standard optimization techniques, such as quasi-Newton methods, can be employed to minimize J_{SE}. Since this optimization is done in a low-dimensional space (P), it can be done very fast. Empirically, learning such a weak classifier is faster than learning a domain-partitioning classifier using dynamic programming, especially

when the degree is moderately large (e.g. $P = 10$).

The classifier (4.25) involves a logistic regression–style posterior, whose coefficients are learned using a technique similar to LR but with a modified cost function. Hence it is called the Quasi Logistic Regression weak classifier. As we have suggested for the domain-partitioning weak classifier, an affine transform can be applied to (4.25), somewhat blending Gentle AdaBoost with Real AdaBoost.

4.3 BooMDA: Boosted Maximal Divergence Analysis

BooMDA is a boosting classifier aggregating MDA weak classifiers under the Gentle AdaBoost framework. So far we have discussed all the components in BooMDA; some preliminary experimental results achieved by this algorithm are shown in this section. In Figure 4.4 we show the decision boundary learned by BooMDA in 40 iterations with 3rd degree quasi LR weak classifiers.

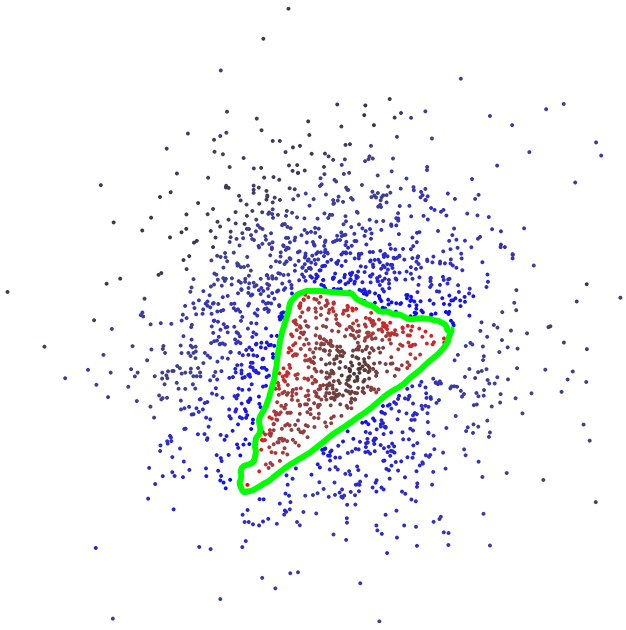


Figure 4.4: Triangle example: Classifier learned by BooMDA in 40 iterations, using 3rd degree quasi LR weak classifiers.

Figure 4.5 shows a more complicated spiral example. The decision boundary was learned in 100 iterations with 10th degree quasi LR weak classifiers.

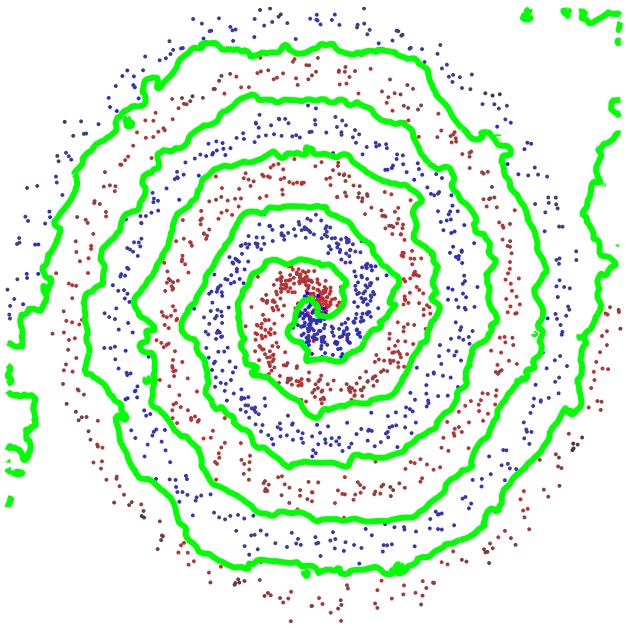


Figure 4.5: Spiral example: Classifier learned by BooMDA in 100 iterations, using 10th degree quasi LR weak classifiers.

4.4 Summary

BooMDA is closely related to the approaches we discussed in earlier chapters, as well as several other boosting classifiers. Here we make a comparison to these relevant approaches.

4.4.1 MDA vs. SODA

Both MDA and SODA aim at learning a discriminative projection vector under a Gaussian assumption. In SODA we need to compute three features separately and select from them the best for classification. On the other hand, MDA learns the projection in one shot, obtaining the feature that is consistently better, or at least as good as, the best of SODA.

Formulated explicitly as an optimization problem, the MDA algorithm is able to include regularization terms, favoring solutions with certain attributes. For example, when the L^1 term is added, sparsity constraint can be enforced to the projection vector. On the other hand, it is not as easy to apply such regularization to SODA (FLD and MRC).

SODA has a closed form solution, while MDA requires numerical optimization, which (1) is less efficient; and (2) has the local minimum problem. However, when we perform MDA in BooMDA, we can use the result from the last iteration as initialization, reducing the overall computational cost. On the other hand, MDA only needs to compute one projection vector, while straightforward implementation of MRC using off-the-shelf numerical libraries usually involves computing all generalized eigenvectors. The speeds of these two methods need to be compared in the future. For the local minimum problem, in BooMDA we may employ a "cold-starting" strategy, using SODA to initialize MDA every several iterations, or when we find that BooMDA is stuck with a single MDA feature.

4.4.2 MDA vs. KLA

KLA is the projection pursuit approach proposed in [21], maximizing the Kullback-Leibler divergence between two classes. Neither KLA nor MDA has a closed form solution and iterative optimization has to be employed to seek the projection vector in both. However, MDA is more efficient in two ways: (1) It works on the covariance matrices of the two classes, which is the "summary" of the training data. Using such summarized representation is more efficient, especially when there is a huge number of training samples, so MDA is much more scalable. As a specific case, in face recognition MDA can work in a similar fashion as MRC-Boosting to handle a large number of training faces. (2) The objective function is much easier to optimize. In KLA the cost function is defined using the

training samples in a nonparametric way. The cost function is not differentiable and hard to minimize. In [21] an ad hoc, data-driven optimization algorithm was used. In MDA, we have a smoother, differentiable cost function, and can employ a principled numerical optimization approach.

4.4.3 Two-stage strategy to learn weak classifiers

Like our earlier approaches (MRC-Boosting and SODA-Boosting), BooMDA *decouples* weak classifier learning into two steps: feature learning and classification function learning, which greatly simplifies the learning task. In the first step, we assume Gaussian distribution for two classes in order to formulate a feasible minimization problem in a high-dimensional space; in the second step, we relax the Gaussian assumption so as to find a good discriminant function. Although seeking a discriminant function involves a more complicated cost function, the optimization is in a low-dimensional space, and hence still can be solved effectively.

Generally speaking, BooMDA can be regarded as a refined version of SODA-Boosting. It shares similarities with the latter, but is more principled, and generalizes the latter in different ways (i.e., in both feature pursuit and classifier learning). Naturally, it is quite interesting to apply this algorithm to biometric problems, and study its performance in practice, which is left for future investigation. So far we have presented several related classification algorithms, which can serve as the recognition module in a face processing system. Starting from the next chapter, we will move to the alignment stage, which bridges face detection and recognition, and investigate appearance-based algorithms for that task.

CHAPTER 5

FACE ALIGNMENT WITH LINEAR APPEARANCE MODEL

As we mentioned in Chapter 1, typical automatic facial processing systems consist of three major modules: detection, alignment and recognition. We bypass the detection stage, since it is among the relatively well-solved problems in the whole computer vision community. So far, we have discussed only the recognition stage; in previous chapters, we assumed that the faces to be recognized, as well as those used in the training phase, are already aligned. In this chapter, we discuss the alignment problem, and present a few appearance-based solutions developed for and used in our face processing systems.

In the most general sense, face *alignment* means removing (or normalizing) the unwanted variations in the recognition stage, such as location, viewpoint, illumination, or even expression. In the narrower sense, alignment usually specifically refers to normalizing faces with respect to *location*, since location variability is most commonly met in practice, and if location cannot be well normalized, one could not expect appearance-based recognition algorithms to work robustly. In this chapter we employ this narrower-sense definition.

It should be noted that human faces are "flexible" patterns. Although faces belonging to different people are generally similar, their subtle shapes vary considerably. As a result, to align two faces exactly we need to localize many important facial feature points. Many algorithms have been developed for this purpose, including the Active Shape Model (ASM) [4], Active Appearance Model (AAM) [5], and Linear Morphable Model [6]. We categorize these methods as

"feature-based", and call the alignment they can achieve *strong alignment*.

On the other hand, appearance-based approaches for face alignment do not estimate the exact locations of facial features. They provide an accurate estimation of the face's overall, or global, location, e.g., translation, in-plane-rotation and scaling. With this information, we may extract face images with normalized locations, which can be readily fed into the recognition stage. We call the alignment achieved by appearance-based approaches *weak alignment*.

It should be clarified that *weak alignment* is not necessarily "weak." First, with the location information provided by such alignment, we can effectively remove the location variability of the face to be recognized. The results presented in the previous chapters show that recognition tasks can be reliably conducted for face images processed in this way. Moreover, in certain scenarios, weak alignment is even advantageous. For instance, when the input images are in low resolution, localizing facial feature points becomes unstable, if even possible. Hence the feature-based approaches may not work robustly, whereas weak alignment by appearance-based approaches can still be done.

In this chapter we discuss appearance-based algorithms for aligning face images. A weak alignment algorithm based on linear appearance model is presented, which can reliably and rapidly align face images of a fixed view (say, frontal). Besides working well in a real-time recognition system described later in this chapter, the importance of this alignment algorithm lies in the analysis-by-synthesis paradigm it employed. In Chapter 7, we shall extend it to a more powerful alignment algorithm, which effectively handles varying viewpoints and illumination conditions, employing the same paradigm. In this chapter we also propose an approach capable of automatically aligning a whole ensemble of face images concurrently, built upon the linear model–based alignment algorithm. With such an algorithm, one can build a face alignment model from a set of unaligned face images, waiving the need for

manual alignment, which is a costly and tedious procedure.

5.1 Linear Appearance Model of the Human Face

A linear model is probably the simplest and most widely used model for the appearance of the human face. Despite its simplicity, a linear model is able to capture many appearance variations, especially when the pose of faces is fixed (e.g. frontal). As early as around 1990, the Eigenfaces model [8, 9] was proposed for face recognition, implying the linear model's capability of modeling the differences between faces belonging to distinct people. Study on the effect of illumination has revealed that the appearance change due to different illumination conditions can be accurately modeled by a linear model [45, 46, 47]. The major limitation of the linear appearance model is that it cannot well model the appearance variation due to viewpoint change. Now we focus on modeling the appearance of faces in fixed view (say, frontal) which is a common assumption in many practical applications and most existing work on face processing.

With a standard Eigenfaces model, we model the appearance of faces as

$$\tilde{\mathbf{T}} = \mathbf{T}_0 + \mathbf{U}\tilde{\mathbf{a}}, \tag{5.1}$$

where $\mathbf{T}_0$ is the mean appearance, $\mathbf{U}$ is the matrix consisting of the bases (i.e., Eigenfaces), and $\tilde{\mathbf{a}}$ is the coefficients. To model the possible change in global luminance (due to camera gain control, for instance), we add an affine transform to the appearance

$$\mathbf{T} = g\tilde{\mathbf{T}} + b, \tag{5.2}$$

where g is the gain and b is the offset. Combining (5.1) and (5.2) leads to a

homogeneous linear model for appearance

$$\mathbf{T} = \mathbf{B}\mathbf{a}, \tag{5.3}$$

where the basis matrix $\mathbf{B} = \begin{bmatrix} \mathbf{T}_0 & \mathbf{U} & \mathbf{1} \end{bmatrix}$ ($\mathbf{1}$ being a vector consisting of all ones), and the coefficient vector $\mathbf{a} = \begin{bmatrix} g & g\tilde{\mathbf{a}}^T & b \end{bmatrix}^T$.

5.2 EigenAlign: Face Alignment with Linear Appearance Model

With the linear model (5.3), face alignment can be done by fitting the model to the input image $\mathcal{I}$. Specifically, we need to find the location of the face, parameterized as $\mathbf{p}$, as well as a set of appearance coefficients $\mathbf{a}$, so that the face synthesized by (5.3) best matches the input. In other words, face alignment is achieved in an analysis-by-synthesis fashion. Formally, this is modeled as a minimization problem

$$(\mathbf{p}^*, \mathbf{a}^*) = \min_{\mathbf{p},\mathbf{a}} J \tag{5.4}$$

with the cost function

$$J(\mathbf{p}, \mathbf{a}) = \tfrac{1}{2} \left\| \mathbf{B}\mathbf{a} - \mathbf{I}(\mathbf{p}) \right\|^2, \tag{5.5}$$

where $\mathbf{I}(\mathbf{p})$ refers to the image patch extracted within the input $\mathcal{I}$ at location $\mathbf{p}$. The k-th pixel of $\mathbf{I}$ corresponds to a pixel in the input image $\mathcal{I}$, formally

$$\mathbf{I}_k(\mathbf{p}) = \mathcal{I}\left[\mathbf{W}(\mathbf{x}_k; \mathbf{p})\right], \tag{5.6}$$

where $\mathbf{x}_k$ is the coordinates of pixel k within the template frame, i.e. the frame of the face images represented by our linear model, and warping operator $\mathbf{W}(\cdot; \mathbf{p})$

maps coordinates in the template frame into the input frame (the frame of input image $\mathcal{I}$).

As cost function (5.5) is in the form of the sum of squares, minimization in (5.4) can be done via standard nonlinear LS techniques, such as the Gauss-Newton or Levenberg-Marquardt algorithm. To apply the Gauss-Newton method, we write the Jacobian of (5.5):

$$\mathbf{J} = \begin{bmatrix} \mathbf{J_p} & \mathbf{J_a} \end{bmatrix} = \begin{bmatrix} -\frac{\partial \mathbf{I}}{\partial \mathbf{p}} & \mathbf{B} \end{bmatrix}, \tag{5.7}$$

where $\frac{\partial \mathbf{I}}{\partial \mathbf{p}} = \begin{bmatrix} \frac{\partial \mathbf{I}_1}{\partial \mathbf{p}} & \dots & \frac{\partial \mathbf{I}_N}{\partial \mathbf{p}} \end{bmatrix}^T$ is the Jacobian of the warped and vectorized image $\mathbf{I}$. We have

$$\frac{\partial \mathbf{I}_k}{\partial \mathbf{p}} = \left.\frac{\partial \mathcal{I}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{W}(\mathbf{x}_k;\mathbf{p})} \left.\frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right|_{\mathbf{x}_k,\mathbf{p}} \tag{5.8}$$

where $\frac{\partial \mathcal{I}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathcal{I}}{\partial x} & \frac{\partial \mathcal{I}}{\partial y} \end{bmatrix}$ is the image gradients computed within the frame of input image $\mathcal{I}$. If $\mathbf{W}(\cdot;\mathbf{p})$ is a similarity transformation, $\mathbf{p}$ can be parameterized as $\mathbf{p} = \begin{bmatrix} t_x & t_y & s_x & s_y \end{bmatrix}^T$ with (t_x, t_y) being translation and $s_x = s \cdot \cos(\theta)$ and $s_y = s \cdot \sin(\theta)$ re-parameterizing the scale factor s and rotation angle θ. In this case,

$$\left.\frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right|_{\mathbf{x}_k} = \begin{bmatrix} 1 & 0 & x_k & -y_k \\ 0 & 1 & y_k & x_k \end{bmatrix}. \tag{5.9}$$

The Gauss-Newton update is obtained by solving the linear LS problem locally

$$\min_{\Delta\mathbf{p},\Delta\mathbf{a}} \|\mathbf{B}\mathbf{a} - \mathbf{I} + \mathbf{J_p}\Delta\mathbf{p} + \mathbf{J_a}\Delta\mathbf{a}\|^2.$$

In other words, it is the LS solution to an overdetermined linear system

$$\begin{bmatrix} \frac{\partial \mathbf{I}}{\partial \mathbf{p}} & -B \end{bmatrix} \begin{bmatrix} \Delta \mathbf{p} \\ \Delta \mathbf{a} \end{bmatrix} = \mathbf{Ba} - \mathbf{I} \tag{5.10}$$

Alternatively, general optimization methods such as quasi-Newton or conjugate gradient can be employed. To this end, the gradients of (5.5) can be written as follows:

$$\nabla_{\mathbf{p}} J = -\left(\frac{\partial \mathbf{I}}{\partial \mathbf{p}}\right)^T [\mathbf{Ba} - \mathbf{I}] \tag{5.11}$$

$$\nabla_{\mathbf{a}} J = \mathbf{B}^T [\mathbf{Ba} - \mathbf{I}] . \tag{5.12}$$

One advantage of employing general optimization methods is that we may add to the cost function (5.5) regularization terms that are not in the form of sum-of-squares. We shall see in Chapter 7 how such a regularization technique may become helpful, although it is not employed here.

5.2.1 Multi-resolution alignment

Multi-resolution (hierarchical, or coarse-to-fine) techniques have been proven to be very effective in motion estimation [48]. The recent result, due to Brox et al. [49], revealed its relationship to a principled numerical optimization scheme. This technique is very helpful in our case as well, especially when the initial estimation of the face location is far from the optimum.

In our problem we fit an appearance model to an input image. To implement multi-resolution alignment, a pyramid must be built for both the model and the image, which is done by a series of blurring and down-sampling. For the input image this is not really a burden. However, for the appearance model, building a hierarchical version implies that we have to build a pyramid for each of the training

images and train a separate appearance model at each resolution. To achieve more efficient coarse-to-fine alignment that preserves the advantage of the full multi-resolution scheme, in our implementation we made a few simplifications:

1. We only keep a single appearance model, without building a pyramid for it.

2. We only blur the input image, without down-sampling it. In this way we obtain a series of images $\{\mathcal{I}_l, l = 0, 1, \ldots, L\}$ where $\mathcal{I}_0 = \mathcal{I}$ and $\mathcal{I}_k = \mathcal{I}_{k-1} * \mathcal{G}$ with $\mathcal{G}$ being a blurring kernel (e.g. Gaussian). Starting from $\mathcal{I}_L$ which is the most blurry, or coarsest, image, at each resolution we perform alignment and carry the resulting location as the initialization for the finer resolution.

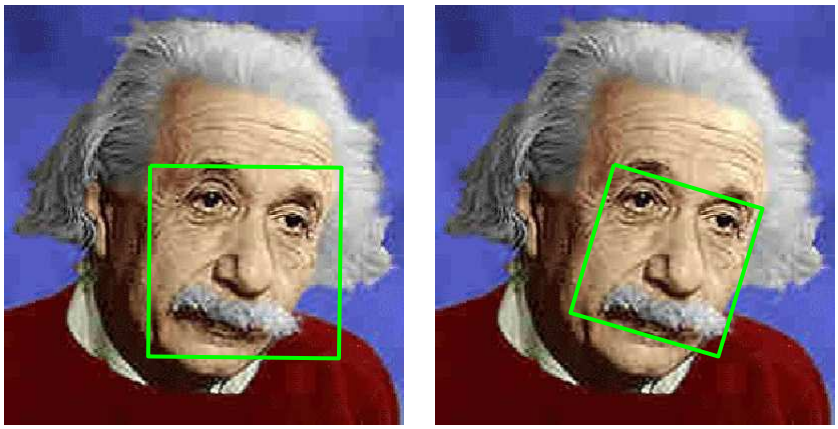
Blurring the input image has the effect of making the cost function smoother; in this sense the strategy taken here may be considered analogous to a deterministic annealing approach, which helps prevent an optimizer from getting trapped in local minima. We found that this simplified coarse-to-fine technique works well in practice and achieves accurate alignment results even when the initial location is relatively far from the optimum. Meanwhile, it accelerates convergence speed compared to single-resolution alignment.

5.3 Results

We built the linear appearance model from 8000 prealigned (by manually marking the eye corners) face images. This is simply done by learning the Eigenfaces model (5.1) of these images. In Figure 5.1 we display the mean face $\mathbf{T}_0$ and the first 9 Eigenfaces. It can be seen that some of the Eigenfaces reflect the effect of illumination.

Then we perform face alignment with this linear appearance model, using the first 20 Eigenfaces. Figure 5.2 shows the alignment result of an example image.

Figure 5.1: Eigenfaces learned from 8000 prealigned faces of the YGA database.



(a) Initial location (b) Aligned location

(c) Extracted face at the aligned location (d) Appearance model fit to 5.2(c)

Figure 5.2: Face alignment using linear appearance model.

Note that the face to fit is not in precise frontal view, whereas our model is built from purely frontal-view faces. This indicates that face alignment with linear appearance model does tolerate, to some extent, out-of-plane rotation of faces.

5.3.1 Application: real-time gender recognition

As we mentioned, in most automatic face processing systems, face alignment is a significant module before face images are fed to recognition algorithms. We implemented a fully automatic, real-time gender recognition system for frontal-view

faces. The linear appearance model–based alignment algorithm presented in this chapter constitutes the alignment module, while the recognition module is built with the SODA-Boosting algorithm discussed in Chapter 3. The OpenCV [2] frontal face detector is the frontmost module detecting faces. Due to the efficiency of all these modules, the system runs in real-time at about 5 frames per second on a typical laptop PC (Pentium M at 1.4GHz). A screen shot of the system is shown in Figure 5.3.

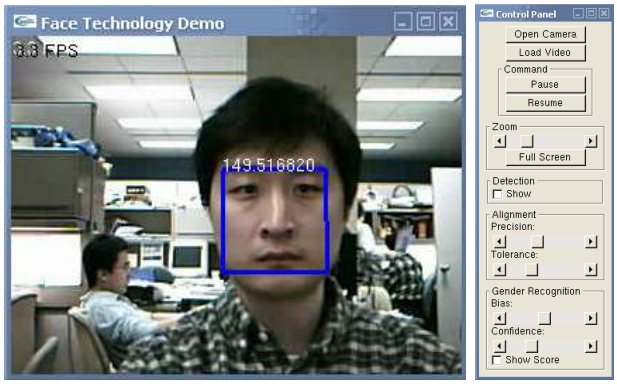


Figure 5.3: Real-time gender recognition system with linear appearance model–based face alignment.

5.4 EnsembleAlign: Aligning an Ensemble of Images via Bootstrapping

Aligning faces using the algorithm discussed so far requires, of course, a ready-to-use linear model. Building such a model requires a collection of face images where the faces are already aligned. One common practice is to label the location of a few feature points (e.g. eye corners, the nose tip etc.) manually in every training image, then normalize the position of the faces with the help of these landmarks. When there are a large number of training images, this is clearly a costly and tedious procedure. In this section, we present a simple technique to automate this task. The key idea is to "bootstrap" the alignment procedure,

starting from a model trained on unaligned (to be precise, only *roughly aligned*) face images, and refining the model iteratively. It is outlined as Algorithm 5.1.

Algorithm 5.1: EnsembleAlign algorithm

Input: Image ensemble $\{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_N\}$, rough initial alignment $\left\{\mathbf{p}_1^{(0)}, \mathbf{p}_2^{(0)}, \ldots, \mathbf{p}_N^{(0)}\right\}$

for $t = 1, 2, \ldots, T$ **do**

Train an Eigenfaces model (5.1), and build linear appearance model (5.3).

Align $\{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_N\}$ with the current model and initialization $\left\{\mathbf{p}_1^{(t-1)}, \mathbf{p}_2^{(t-1)}, \ldots, \mathbf{p}_N^{(t-1)}\right\}$, resulting in new alignment $\left\{\mathbf{p}_1^{(t)}, \mathbf{p}_2^{(t)}, \ldots, \mathbf{p}_N^{(t)}\right\}$

Anti-drifting: Apply a global similarity transformation $\mathbf{p}$ to $\left\{\mathbf{p}_1^{(t)}, \mathbf{p}_2^{(t)}, \ldots, \mathbf{p}_N^{(t)}\right\}$ so that their average transformation is identical to that of $\left\{\mathbf{p}_1^{(t-1)}, \mathbf{p}_2^{(t-1)}, \ldots, \mathbf{p}_N^{(t-1)}\right\}$.

end

Output: Final alignment $\left\{\mathbf{p}_1^{(T)}, \mathbf{p}_2^{(T)}, \ldots, \mathbf{p}_N^{(T)}\right\}$.

5.4.1 Experimental results

In the experiment, we selected from the CAS-PEAL-R1 database [50] 4687 images of frontal faces under varying illumination. The initial face location was provided by the PittPatt face detector [3], as shown in Figure 5.4(a). It can be observed that although the face detector gives the rough location, the faces are by no means well aligned.

The face images at the 50th iteration of the EnsembleAlign algorithm are shown in Figure 5.4(b). The face images have been concurrently normalized with respect to location. Note that even the faces with rather poor initial alignment can also be correctly aligned to other faces.

(a) Sample face images at initial location

(b) Face images at the 50th iteration of the EnsembleAlign algorithm

Figure 5.4: Concurrently aligning an ensemble of face images via the EnsembleAlign algorithm.

5.5 Summary

In this chapter, we introduced two practical appearance-based face alignment algorithms. The EigenAlign algorithm is based on a linear model of face appearance, and is able to reliably and rapidly align faces in approximately a fixed view (say, frontal). The algorithm was successfully employed in our real-time gender recognition system. Another algorithm, EnsembleAlign, is a meta-algorithm aiming at simultaneously aligning a large number of face images that have only been roughly aligned (e.g., from the result of a face detector). The algorithm works in a bootstrapping way, iteratively aligning the faces and retraining the alignment algorithm. It works for appearance-based face alignment algorithms that do not need precisely aligned images for training, for which EigenAlign is a good example. Experimental results were provided to demonstrate the capability of the introduced algorithms.

Although the algorithms introduced in this chapter are designed to work for a fixed view, the basic principle and techniques used here are quite general, and can

be extended to handle cases where multiple views and complicated illuminations are involved. We will cover such an extension in Chapter 7.

CHAPTER 6

MULTIPLE VIEW FACE PROCESSING

In the previous chapters we have limited our discussion to frontal-view face images. Indeed, in most existing work on face processing, researchers have been focusing on the frontal-view case. For example, earlier works on gender recognition have been only able to handle frontal faces images [32, 33, 36, 37]. In fact, most reported experiments were carried out on the FERET database [24], where imaging conditions are well controlled: frontal face, normal illumination, clean background and good image quality. Although some other work reported results on nonfrontal faces [34], the database they used consists of only a small number (100 male and 100 female) of images synthesized using a 3D face database; thus, the imaging condition is over-controlled, i.e., clean background, ideal lighting, high resolution and no imaging noise. However, in many practical applications, especially surveillance or Electronic Customer Relationship Management (ECRM, e.g., automatically collecting customers' demographic data through a camera), image quality seldom reaches such an ideal level. The images captured from real-world videos usually, if not always, are of rather poor quality, which means low resolution, arbitrary illumination, imaging noises, and most importantly, the faces are often captured in a nonfrontal view.

As a result, our current and future research aims at extending our current face processing capability to face images of multiple views, which is the topic of this chapter. First we discuss four general strategies to accommodate multiple-view face images in recognition tasks: universal, view-adaptive, normalization-based and

hybrid. Then we present experimental results on multiple-view gender recognition along the paths of view-adaptive and universal approaches, therefore gaining an insight into their advantages and disadvantages.

6.1 Strategies for Multiple View Recognition

Generally speaking, to perform recognition tasks for face images in different views, it is natural to reduce the problem to the simpler single-view case, so that existing algorithms can be applied indirectly. To this end, there are several potential approaches:

- **Universal approach.** View variation is ignored, and images across all the views are treated equally. In the training stage, face images in different views are used all together to train a "universal" classifier. In the recognition stage, the novel face is classified by this "universal classifier" to reach to decision. This is a "black-box" approach, where the intrinsic characteristics of the samples are hidden from the classifier. The advantage of this approach is that no viewpoint estimation needs to be done in either the training stage or the recognition stage.

- **View-adaptive approach.** The recognition algorithm is trained for each view independently. To make this approach computationally feasible, one needs to quantize the view space into N ranges, where N cannot be too large (e.g., 50). In the recognition stage, first the viewpoint of the novel face should be estimated, then the corresponding classifier for that view is chosen to do the recognition. Compared to the "universal classifier," the view-adaptive classifiers may have a higher recognition accuracy, as the underlying classification problem is easier than the case where view variation is involved.

However, an additional view estimation step is required in the recognition stage.

- **Normalization approach.** The nonfrontal face image is warped to a frontal-view face image. Then, frontal-view facial recognition algorithms can be applied. When the view departs only slightly from frontal, a 2D transformation, such as an affine transform, can be sufficient; otherwise a 3D model–based approach may be necessary (which is potentially computationally expensive.)

- **Hybrid approach.** Finally we envision a hybrid approach, where "mild" warping is used to keep N, the number of quantized views, relatively small. A potential procedure is as follows: The view space is divided into N ranges. For each range, a representative view is chosen. When a test nonfrontal-view face image is encountered, we first estimate/recognize which range-bin it falls in. Then the test image is warped to the representative view of this range. As the novel view does not depart much from the representative view, fast 2D transformation suffices. The recognition algorithm for that particular representative view is used to carry out the recognition task. (One recognition algorithm was trained for each of the N representative views.)

Employing one of these strategies, we may apply algorithms developed to recognize face images of a fixed view to handle multiple-view problems. It should be noticed that if the recognition algorithm can handle not only image input, but also other representations of the human face, there is another possibility, where normalization is done "implicitly." That is, the frontal-view face images are not explicitly synthesized; instead a *view-independent representation* is obtained. This strategy was employed in face recognition using the 3D morphable model [13].

Among all these approaches, a normalization approach almost surely requires the help of a 3D model, and in the fitting of a 3D model, a pointwise correspondence between the input image and the model usually needs to be built. In fact, state-of-the-art 3D face analysis approaches [51, 52, 53] all fall into the category of feature-based approaches. The hybrid approach sits somewhere between the view-adaptive and normalization approaches; although it generally does not require fitting 3D models, its warping step implicitly involves pixel-level correspondence, setting it apart from appearance-based approaches. On the other hand, the first two strategies, view-adaptive and universal, are most relevant for appearance-based face processing, as they need to deal only with holistic appearance of face images. Therefore, they are of particular interest within this dissertation's main theme. The next two sections are dedicated to the two approaches, presenting experimental results with them.

6.2 View-Adaptive Gender Recognition

In this section, we present experimental results of multiple-view gender recognition following the line of view-adaptive approach.

In order to study gender recognition from multiple views, one important task is to obtain a suitable face database involving large view variation. The face database should contain reasonably many people so that experiments on it (through some sort of cross-validation) do help us know an algorithm's generalization performance. The face images should be labeled (both view and gender in our problem) and aligned. Unfortunately, there is no such good database that is publicly accessible. Therefore as a starting point we chose to synthesize images from 3D models. We collected more than 400 3D faces using a Cyberware 3D scanner, which are in neutral expression, are well balanced between male and female, and contain various

ethnic groups (Caucasian, African, South and East Asian) [54]. We also obtained about 200 scans from the University of South Florida. Finally we composed a 3D face database consisting of 675 people, from which face images of different views under variable illumination conditions can be directly synthesized. It is worth mentioning that on all the 3D scans major facial features (such as eye corners, nose tip and mouth corners) were manually labeled, hence the 2D location of the rendered faces can be precisely controlled. In this way, we have rendered face images in 9 different views, namely $0°$, $\pm 15°$, $\pm 30°$, $\pm 36°$ and $\pm 45°$, in pan angle, as show in Figure 6.1. Currently we do not introduce varying illumination, in order to focus on the effects due to view change only.

Figure 6.1: Face images in different views rendered from UIUC 3D face database.

First experiment studied the effect of view change on the accuracy of gender recognition, where for each of the 9 views, we separately performed gender recognition with SVM (of RBF kernel). The protocol was similar to that in the last chapter. In each run of experiments, we randomly partition the images into training and test set. Each time, 80% were selected for training and the remaining images were used for test. Note that for each view every person appears exactly once; hence the people in the training and testing sets do not overlap. The recognition

accuracy (mean and standard deviation) is plotted in Figure 6.2. It can be observed that the accuracy for views apart from frontal is lower than for the frontal case; the performance drop is not significant (about 3%) though. However, it should be noted that such small performance difference is partly due to the completely clean background of the synthesized images. In real images of nonfrontal-view faces, the inclusion of *unpredictable* background region introduces considerable difficulty for recognition and needs further study.

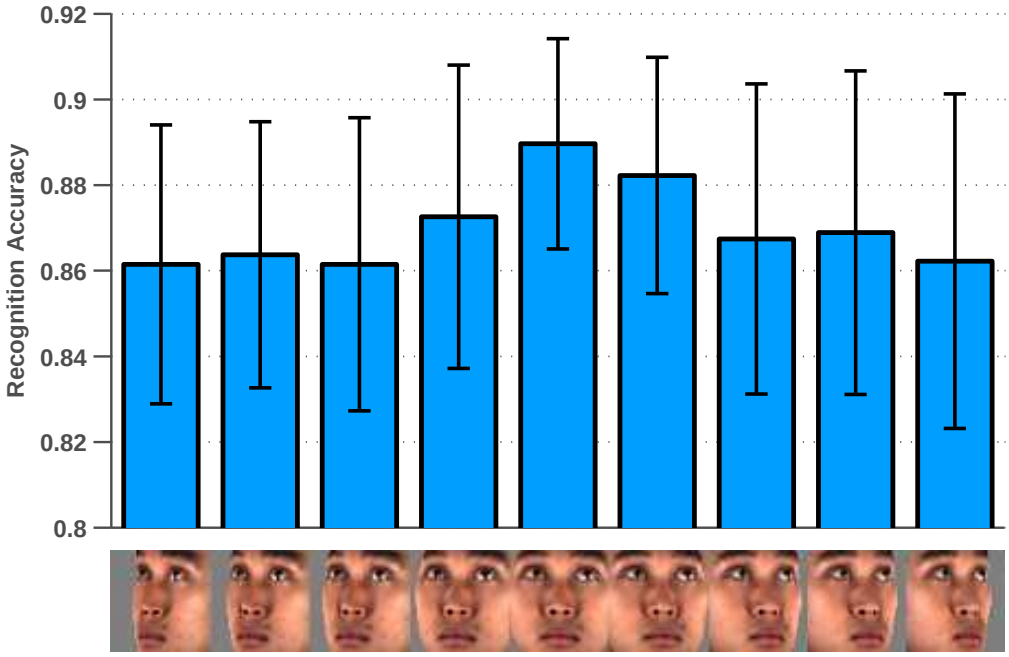


Figure 6.2: Gender recognition accuracy for different views.

The above experiment simulates the ideal case that the view of the new face can be accurately estimated so that the recognizer for that specific view can be correctly selected to do the recognition. In practice, we of course cannot guarantee, or even expect, the view estimator to work perfectly. Therefore it is of much interest to study how the recognition accuracy degenerates when the recognizer for a certain view is applied to faces of another view. To this end, we designed another group of experiments, where we trained a gender recognizer (again, SVM) for each of the 9

views, and tested it on other views. If the previous experiment is also included, we conducted a total of 9×9 experiments. In each of these experiments, the protocol was the same as before, and the mean accuracy was recorded. The results are shown in Figure 6.3. Apparently the degeneration due to incorrect view estimation does not seem to be disastrous, as the worst accuracy was still about 70%, which implies that gender recognition for multiple views is feasible even when we do not have a very accurate view estimator. However, again it should be pointed out that in realistic scenarios the degeneration will certainly be more significant because of the adverse effect of background and other noises.

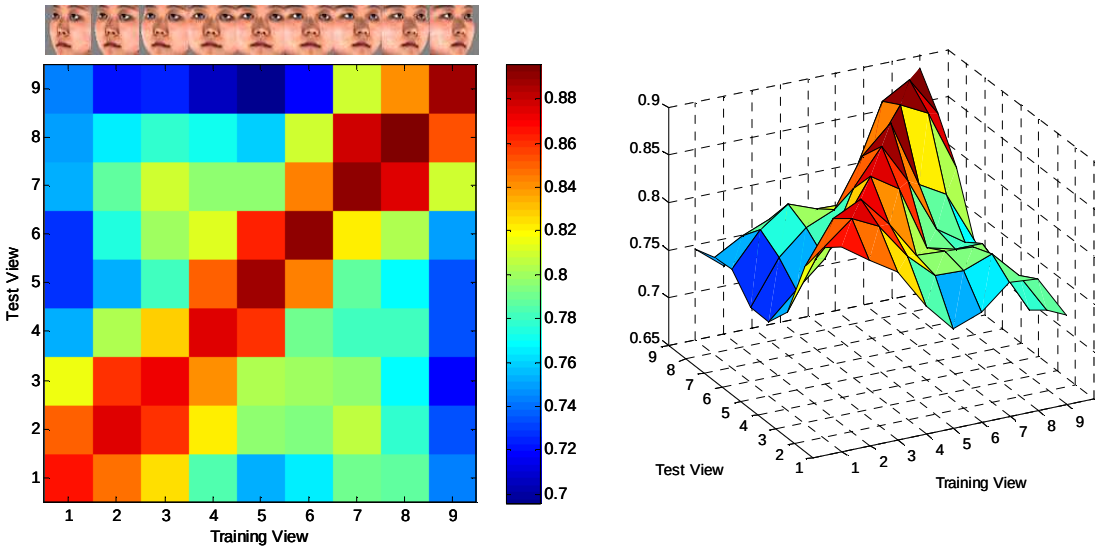


Figure 6.3: Gender recognition crossing different views.

6.3 Multiple-View Gender Recognition with Universal Strategy

The view-adaptive gender recognition we just studied inevitably requires a view estimation module, whose accuracy directly affects the final recognition accuracy. Although we shall come back to the problem of view estimation in the Chapter 7,

for now it is worth looking into the universal strategy, since it entirely waives this step, providing an alternative solution. The cost of bypassing the extra task is that the classification problem posed to the recognition module is clearly harder than in the view-adaptive case. The hope is that if the recognizer is strong enough to accommodate the harder problem, achieving reasonably good accuracy, it will also be a promising way to tackle multi-view facial recognition problems.

To this end, we conducted a new set of experiments evaluating the universal approach, on the same database. As before, 80% face images of different views are randomly selected for training, and the remaining 20% for test. What differs from the previous experiment is that in the training stage, images from *all* views are used to train a universal classifier, while in the recognition stage the test images of different views, which were not seen in training, are all classfied with the classifier. As before, we recorded the avarage accuracy and the standard deviation from 10 random runs, as shown in Figure 6.4; for comparison the accuracy of the view-adaptive approach is also plotted.

Comparing the accuracy of the universal approach with that of the view-adaptive approach, it is clearly observed that the universal approach *uniformly* surpasses the latter. It is quite interesting to notice such a performance gain. Because the universal approach has to tackle a harder classification problem, so intuitively one would expect it to have lower accuracy. One possible reason for this seemingly contradicting observation may be the limited size of our dataset. In training a universal classifier, many more training samples are used, compared to the training of view-adaptive classifiers. For any specific view, while training the universal classifier the samples from similar views supply additional information, improving the stability of the trained classifier. However, this may also suggest that classification problems for different views, especially close views, are relevant. Therefore, solving them as a whole provides an opportunity for them to interact

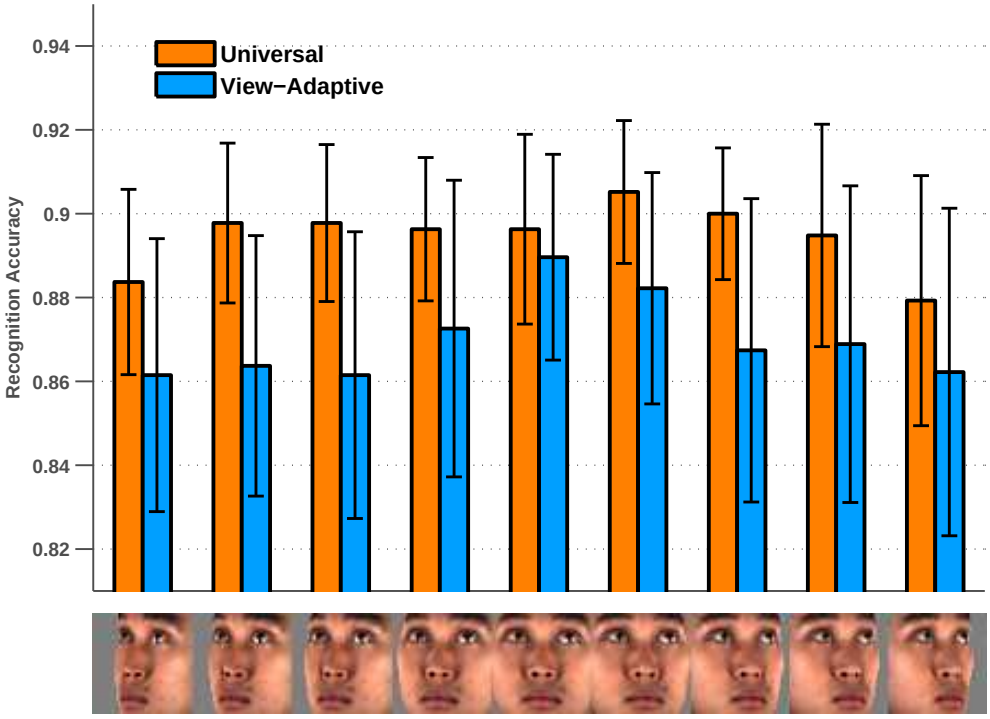


Figure 6.4: Gender recognition accuracy for different views: universal vs. view-adaptive.

and help each other.

6.4 Summary

In this chapter, we presented general strategies to extend the capability of frontal face processing techniques to multi-view scenario. Among the four approaches, view-adaptive and universal are most related to appearance-based face processing; therefore, experiments were conducted to discover their advantages and disadvantages. Due to the lack of a large, publicly available, and labeled database at the time of the studies presented here, we employed a synthetic database rendered from 3D face scans. Although the experiments did discover some knowledge about the studied approaches, to evaluate recognition algorithms in realistic scenarios, a large database of real faces images would be of great value.

Certainly, ultimately we need to step far beyond experiments conducted on preprocessed databases and perform multi-view face analysis on real-world images. Experiments on preprocessed databases, no matter whether synthetic or real images, are considerably simplified compared to a realistic scenario. In these experiments all face images are more or less well-aligned. However, in a realistic face processing system, before recognition can be done, we must (1) determine the view of the input face, and (2) normalize the location variability existing in the face image. These are nontrivial problems, taking into consideration the drastic variability of illumination and viewpoint of real-world face images. In the next chapter, we present an appearance-based approach that throws some light on these problems.

CHAPTER 7

CONCURRENT FACE ALIGNMENT AND ANALYSIS OF VIEW AND ILLUMINATION

Because of the effectiveness of appearance-based approaches in frontal face processing, it is quite appealing to extend them to handle face images of multiple views. As discussed in Chapter 6, view-adaptive and universal strategies may help us, bringing appearance-based recognition techniques to the multi-view scenario. However, applying either strategy, we need to obtain the face appearance in the first place, in a consistent way so that the variability in face location is appropriately normalized. This task is called multi-view alignment, which is a fundamental module in a multi-view face processing system.

Moreover, in many cases, aligning faces is not the only task we need to fulfill in face analysis before the final recognition stage. For example, with view-adaptive approach we need to estimate the view as well. On the other hand, as a fundamental physical factor in the formation of face images, illumination is also of great interest in face analysis. It would be very useful if we could estimate these factors from an input face image. In this chapter, we propose an appearance-based approach that is capable of aligning multi-view faces and estimating their view and illumination at the same time.

The linear appearance model–based alignment we discussed in Chapter 5, although an effective method for frontal-view faces, is not readily applicable to multi-view scenarios. The underlying reason is the high nonlinearity in face appearance when multiple views are involved. Therefore, a more appropriate appearance model needs to be employed in this scenario.

Inspired by results in multilinear algebra [55, 56] and more recently [57, 58], Vasilescu and Terzopoulos [59] introduced the TensorFaces model to the face processing community. The model employs a tensor representation for face images, providing an elegant way to characterize nonlinearly entangled factors involved in the formation of face images, e.g., person identity, view, and illumination, which makes it a promising appearance model for multi-view face images.

In this chapter we propose an appearance-based approach for multi-view face analysis, exploiting a tensor representation of faces. The major difference between our work and the original TensorFaces and following works [59, 60, 61, 62] lies in the fact that those earlier works mainly focused on the tensor concept and the learning issues of tensor models, while how to analyze new face images robustly with the tensor model was relatively understudied. Although [62, 63] introduced an approach along this line, due to its limitations the method has not led to a robust and practical solution to tensor-based face analysis. Our approach attempts to fill this gap, making the tensor appearance model a useful and robust tool for analyzing face images involving complex factors, such as varying view and illumination. As we are going to demonstrate, our approach is capable of estimating view and illumination while concurrently aligning face images to a tensor model. The proposed approach may serve a variety of applications, e.g. facial recognition, human-computer interaction (HCI), and computer graphics (e.g., view/light editing).

7.1 Anisotropic Tensor Model

In this section we discuss how to learn a good tensor model of face appearance. We skip the introduction to the basic conceptions about the tensor approach for face modeling. For a detailed discussion, the readers are referred to [59, 60]. Throughout this chapter, tensors are represented in calligraphic uppercase letters, e.g. $\mathcal{Z}$. As

before, matrices are notated in bold uppercase, and vectors are in bold lowercase, except for variables otherwise described.

In the formation of face images, many different factors, such as person identity (which is characterized by the facial shape and texture), illumination, viewpoint and expression, are entangled with each other in a highly *nonlinear* way. Tensor representation provides an elegant way to separate these factors. Mathematically, the tensor representation is a *multi-linear* model for face images; i.e., it assumes that face appearance is multi-linear with respect to the involved factors, or "modes" in tensor terminology. This certainly extends linear models (e.g. PCA) that have been conventionally used to model face images, and is more competent to model the nonlinear interaction of different factors.

The underlying assumption of the multi-linear model is that when we fix all but one mode we obtain a linear model for that mode. This raises a question: Is a linear model universally appropriate for all modes relevant to face appearance? We would argue that the answer is no. For certain modes, such a linear model is perfectly appropriate. For example, it is well-known that images of a Lambertian object (say the face of a certain person), in fixed view but under varying illumination conditions, can be well approximated by a very low-dimensional linear subspace [46, 64, 47]. For some modes, however, a linear model is disappointingly inappropriate, due to the high *intrinsic nonlinearity* of those modes. For instance, when illumination is fixed, the images of a person's face captured in different views form a highly nonlinear manifold. To see this, imagine that we have images of the same face under the same lighting condition, but seen from different viewpoints, their linear combination may easily lead to blurred images that can no longer be perceived as a face, due to the drastic pixel misalignment introduced by varying views. Finally, some other modes, such as identity and expression, lie somewhere between intrinsically linear and highly nonlinear. This is because pixel

misalignment due to these modes is usually mild, especially for low-resolution face images that are common in appearance-based face processing techniques.

The above discussion suggests that different modes in a tensor model of face images should *not* be treated equivalently. For modes where linearity can be reasonably assumed (including intrinsically linear modes such as illumination or approximately linear modes such as identity), linear modeling is appropriate, but for highly nonlinear modes (e.g., view), we should not try to model them with a simple linear model. In other words, in employing the tensor representation for face analysis, we should take into account the properties of different modes and treat them in an *anisotropic* fashion. For this reason, we refer to a tensor appearance model that respects different characteristics of its modes and treats them differently as an Anisotropic Tensor Model (ATM).

Now we may have a close look at an ATM, showing its advantages in modeling face images. Throughout this chapter, we consider three most commonly encountered modes, view, illumination and person identity. The illustrative examples, as well as experiments coming later, employ face images within the CMU-PIE database [16], which is a publicly available and widely used database involving large, systematically controlled variability in viewpoint and illumination. PIE consists of several subsets focusing on the changes in different factors, and we used the "illum" subset which includes most combinations of different viewpoints and illuminations, best meeting our analysis demands here.

Taking the tensor approach [59], the ensemble of face images involving various formation factors can be represented as a tensor $\mathcal{D}$ where each factor corresponds to a mode. So the PIE face images with changing personal identity, illumination and view are organized as illustrated by Figure 7.1 where subsets of view, illumination and people are shown.

With multilinear algebra, this tensor can be factorized as the multilinear

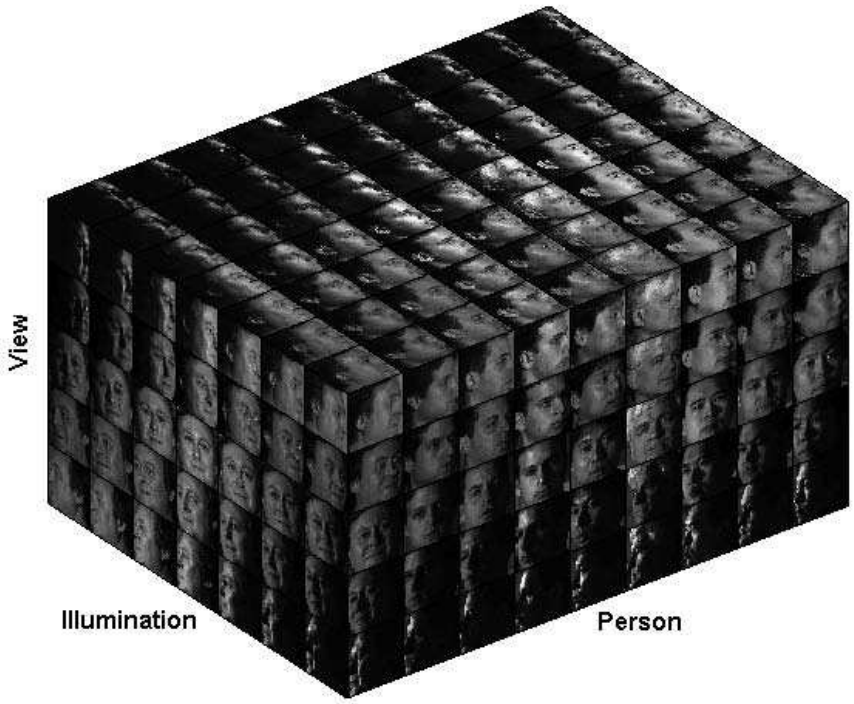


Figure 7.1: Examples of CMU-PIE face images in tensor representation.

multiplication of mode matrices denoted by $\mathbf{U}_{mode}$ and the core tensor $\mathcal{Z}$, which governs the interaction between modes.

$$\mathcal{D} = \mathcal{Z} \times_{pixel} \mathbf{U}_{pixel} \times_{view} \mathbf{U}_{view} \times_{illum} \mathbf{U}_{illum} \times_{ident} \mathbf{U}_{ident} \tag{7.1}$$

Learning a tensor appearance model from the raw tensor data involves multilinear multiplying to it the transposed mode matrices. The TensorFaces appearance model [59] is obtained this way as a tensor shown in (7.2).

$$\mathcal{T} = \mathcal{Z} \times_{pixel} \mathbf{U}_{pixel} \tag{7.2}$$

$$= \mathcal{D} \times_{illum} \mathbf{U}_{illum}^{T} \times_{view} \mathbf{U}_{view}^{T} \times_{ident} \mathbf{U}_{ident}^{T} \tag{7.3}$$

In the construction of the TensorFaces (7.2), it is clear that all relevant modes (except for the pixel mode, which is not a formation factor we need to model) are

treated exactly the same way, or we can say that it is isotropic with respect to modes. Along the identity and illumination modes, linear transformation with the mode matrices is appropriate. Indeed, such transformation is essentially helping us to *compress* the raw data into a more compact model, by preserving only the most significant bases. However, as we have pointed, the view mode should be treated differently. Due to its high nonlinearity, linear modeling along the view mode will destroy its intrinsic structure. As a result, the view mode should be kept untransformed, which leads to an ATM model $\mathcal{B}$:

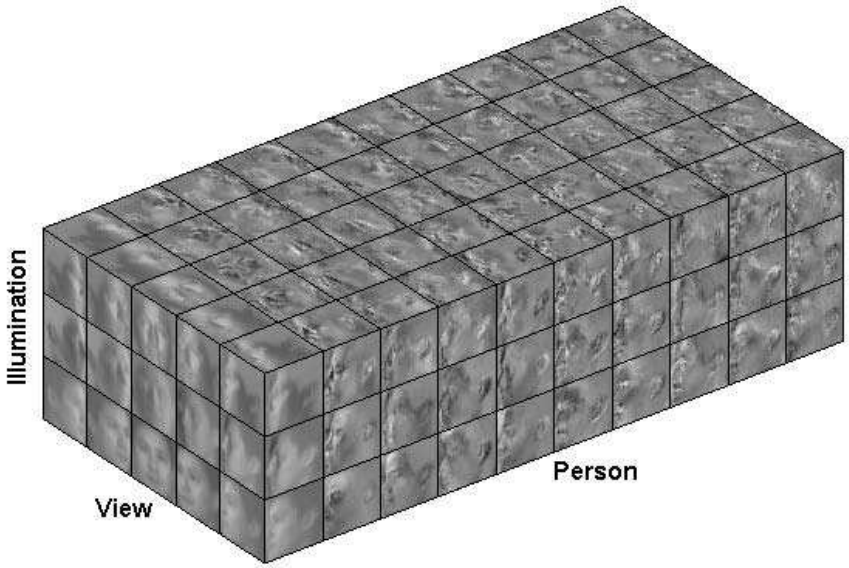
$$\mathcal{B} = \mathcal{Z} \times_{pixel} \mathbf{U}_{pixel} \times_{view} \mathbf{U}_{view} \tag{7.4}$$

$$= \mathcal{D} \times_{illum} \mathbf{U}_{illum}^{T} \times_{ident} \mathbf{U}_{ident}^{T} \tag{7.5}$$

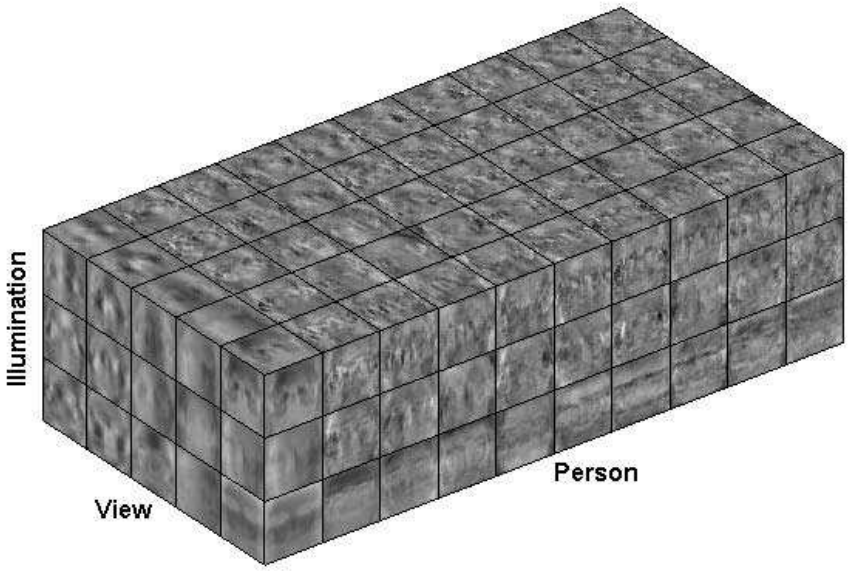
To illustrate the difference between ATM (7.4) and conventional TensorFaces (7.2), the two models constructed from the same raw data (subset of PIE database) are visualized in Figure 7.2.

As demonstrated by Figure 7.2(a), in an ATM the original views are preserved unchanged in the learned model, the bases along the person mode are analogous to the Eigenfaces, while along the illumination mode, it is also easy to observe the bases that compactly characterize the illumination effect (we keep the first three components along the illumination mode, which is able to model the illumination effect of a single light source, as is the case for the PIE dataset we use here). However, in Figure 7.2(b), due to the view mode's nonlinearity, different views mingle with each other, leading to bases that do not preserve original view structure and are not semantically meaningful.

It should be pointed out that, when *no* compression is performed to the view mode, TensorFaces and ATM contain the *same* information, despite the fact that



(a) Anisotropic Tensor Model (ATM)



(b) TensorFaces

Figure 7.2: The difference between ATM and TensorFaces.

the latter is visually more meaningful. However, this is not the only benefit that an ATM representation can bring. In the coming sections, we will show how the preserved view structure can help in face image analysis with the tensor model. On the other hand, as stated before, the true purpose of linear transformation along a certain mode is to compress data along that mode. Due to the intrinsic nonlinearity of the view mode, this is not a reasonable choice, since rough linear approximation would heavily distort the information contained in the raw data. In this sense, keeping the view mode untransformed, as done in ATM, is also sensible.

7.2 Face Analysis with ATM

As with any model-based approach, the application of tensor model to face analysis includes two stages: learning and fitting. We have seen how an ATM was constructed during the learning stage, where a compact tensor representation is learned from the ensemble of training face images undergoing all kinds of variability. In this section the more critical fitting stage is covered, where the learned model is used to analyze *new* face images, obtaining parameters that interpret the new face.

7.2.1 Tensor fitting: An analysis-by-synthesis approach

With the tensor model, any face image can be parameterized by an illumination vector $\mathbf{u}$, a view vector $\mathbf{v}$ as well as an identity vector $\mathbf{w}$, and be synthesized as

$$\mathbf{T}(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \mathcal{B} \times_{illum} \mathbf{u}^T \times_{view} \mathbf{v}^T \times_{ident} \mathbf{w}^T. \tag{7.6}$$

Analyzing a new face image $\mathbf{I}$ (i.e., understanding its identity, illumination, and view) by fitting this tensor model could be done in an analysis-by-synthesis fashion, minimizing

$$F_0(\mathbf{u}, \mathbf{v}, \mathbf{w}, b) = \tfrac{1}{2} \left\| \mathbf{T}(\mathbf{u}, \mathbf{v}, \mathbf{w}) + b - \mathbf{I} \right\|^2, \tag{7.7}$$

which is a nonlinear LS problem. The variable b here is included to model global intensity offset. Unlike the linear appearance model (5.3), where the offset parameter is absorbed into coefficient vector $\mathbf{a}$ by including a constant vector $\mathbf{1}$ in the linear appearance bases, here due to the model's multilinearity, b remains as a separate parameter.

7.2.2 Anisotropy in the fitting stage

As a key characteristic of ATM representation, in the learning stage we obtain a tensor model by leaving the view mode untransformed, so as to preserve its original structure: a discrete sampling of the view manifold. However, if we had multiplied the view mode matrix to the raw data tensor without any effort at compression, the resulted isotropic tensor model would carry the same information as ATM. Therefore, if in the fitting stage we do not respect the special property of the view mode, the ATM is not much different from a regular TensorFaces model. Unfortunately, the cost function (7.7) clearly does not serve this purpose at all, and we need a modification to encode the special property of the view mode.

Recall that in ATM tensor $\mathcal{B}$ views are separate from each other; each slice along the view mode forms a view-based model for that specific view. Also, recall that the view mode is highly nonlinear; hence, when fitting ATM to a new face image, it is undesirable to linearly mix models of many views, which would easily lead to an implausible, blurry fit. Instead a good fit should be limited to merely a few view-specific models. This diagnosis implies that the view vector $\mathbf{v}$ should be highly *sparse*.

To encode the sparse nature of $\mathbf{v}$ we add a regularization term to (7.7), resulting

in

$$F(\mathbf{u},\mathbf{v},\mathbf{w},b;\rho)=\tfrac{1}{2}\left\|\mathbf{T}(\mathbf{u},\mathbf{v},\mathbf{w})+b-\mathbf{I}\right\|^2+\rho R(\mathbf{v}). \tag{7.8}$$

It is well known that the L^1 norm has the property of favoring sparse solutions [65, 66]. However, since we are going to take a nonlinear optimization approach, for which L^1 is not an easy cost function to work with, we choose $L^p\,(1<p<2)$ to enforce sparsity constraint, namely

$$R(\mathbf{v})=\|\mathbf{v}\|_p^p=\sum\nolimits_{i=1}^{V}|\mathbf{v}|^p \tag{7.9}$$

where ρ is the regularization parameter, which determines how sparse the view coefficients will be. The larger ρ is, the sparser will $\mathbf{v}^*$ be.

However, directly minimizing F would simply lead to $\mathbf{v}\approx\mathbf{0}$. This is due to the multi-linear structure of $\mathbf{T}(\cdot)$ whence $\mathbf{v}$ could only be determined up to scaling; as a result the regularization term would bring it scaled down all the way to zero. Therefore we have to fix the scale of $\mathbf{v}$. In addition, one of $\mathbf{u}$ and $\mathbf{w}$ should have fixed scaling as well, to remove unnecessary degree of freedom. To this end, we solve a constrained minimization:

$$(\mathbf{u}^*,\mathbf{v}^*,\mathbf{w}^*,b^*)=\underset{\mathbf{v},\mathbf{u},\mathbf{p},b}{\arg\min}\,F(\mathbf{u},\mathbf{v},\mathbf{w},b)\,\text{s.t.}\;\;\|\mathbf{u}\|=\|\mathbf{v}\|=1 \tag{7.10}$$

The reason why we choose to fix the scale of $\mathbf{u}$ is that usually it has lower dimensionality than $\mathbf{w}$, resulting in less computation in the optimization procedure, as we shall see later.

Now, by introducing the sparsity term for the view mode, we treat it specially in the fitting stage, reflecting the *anisotropic* nature of our approach. The benefit of such treatment shall be clear later in the experiments.

7.3 Riemannian Tensor Fitting

One may immediately consider solving (7.10) with a classic constrained optimization method, such as sequential quadratic programming (SQP). However, it is noticeable that the constraints in (7.10) are very special, i.e. they are simply hyperspheres. It is very appealing to exploit the simple geometric property of the constraints so as to reach a more efficient way to minimize the cost function. Mathematically, (7.10) can be considered as an *unconstrained* optimization problem on Riemannian manifold $\mathcal{M} = \mathcal{S}^{U-1} \times \mathcal{S}^{V-1} \times \mathcal{R}^{W} \times \mathcal{R}$. For this type of problem, one of the most effective approaches is the geometric optimization methods [67, 68, 69], which were developed and have become popular during the last decade. To apply geometric optimization techniques, there are several necessary ingredients that we need: the geodesic, gradient, and Hessian, which we shall cover in this section.

Due to the simple geometry of spheres and Euclidean spaces, the geodesics are given by

$$\mathbf{u}_t = \exp\left(\mathbf{u}, \Delta_{\mathbf{u}} t\right) = \mathbf{u} \cos \|\Delta_{\mathbf{u}}\| \, t + \frac{\Delta_{\mathbf{u}}}{\|\Delta_{\mathbf{u}}\|} \sin \|\Delta_{\mathbf{u}}\| \, t \tag{7.11}$$

$$\mathbf{v}_t = \exp\left(\mathbf{v}, \Delta_{\mathbf{v}} t\right) = \mathbf{v} \cos \|\Delta_{\mathbf{v}}\| \, t + \frac{\Delta_{\mathbf{v}}}{\|\Delta_{\mathbf{v}}\|} \sin \|\Delta_{\mathbf{v}}\| \, t \tag{7.12}$$

$$\mathbf{w}_t = \exp\left(\mathbf{w}, \Delta_{\mathbf{w}} t\right) = \mathbf{w} + \Delta_{\mathbf{w}} t \tag{7.13}$$

$$b_t = \exp\left(b, \Delta_b t\right) = b + \Delta_b t \tag{7.14}$$

The first- and second-order derivatives of F in the direction of $\Delta = [\Delta_{\mathbf{u}}, \Delta_{\mathbf{v}}, \Delta_{\mathbf{w}}, \Delta_b]$ are, respectively,

$$dF\left(\Delta\right) = \left.\frac{dF}{dt}\right|_{t=0} = \nabla_e F^T \left[\begin{array}{cccc} \dot{\mathbf{u}}_t, & \dot{\mathbf{v}}_t, & \dot{\mathbf{w}}_t, & \dot{b}_t \end{array}\right]\Bigg|_{t=0} = \nabla_e F^T \Delta, \tag{7.15}$$

where $\nabla_e F = \left[\begin{array}{cccc} F_{\mathbf{u}}, & F_{\mathbf{v}}, & F_{\mathbf{w}}, & F_b \end{array}\right]$ is the Euclidean gradient; and

$$\text{Hess } F(\Delta,\Delta) = \left.\frac{d^2F}{dt^2}\right|_{t=0} = \left.\frac{d}{dt}\nabla_e F^T \left[\begin{array}{cccc} \dot{\mathbf{u}}_t, & \dot{\mathbf{v}}_t, & \Delta_{\mathbf{w}}, & \Delta_b \end{array}\right]\right|_{t=0} \tag{7.16}$$

$$= \Delta^T \mathbf{H} \Delta + \nabla_e F^T \left[\begin{array}{cccc} \ddot{\mathbf{u}}_t, & \ddot{\mathbf{v}}_t, & \mathbf{0}_W, & 0 \end{array}\right]\Big|_{t=0} \tag{7.17}$$

$$= \Delta^T \mathbf{H} \Delta - \left(F_{\mathbf{u}}^T \mathbf{u}\right) \|\Delta_{\mathbf{u}}\|^2 - \left(F_{\mathbf{v}}^T \mathbf{v}\right) \|\Delta_{\mathbf{v}}\|^2, \tag{7.18}$$

where $\mathbf{H}$ is the Euclidean Hessian matrix. Polarizing Hess $F(\Delta,\Delta)$, we obtain the Hessian function

$$\text{Hess } F(\Delta,\Theta) = \tfrac{1}{4}\left[\text{Hess } F(\Delta+\Theta,\Delta+\Theta) - \text{Hess } F(\Delta-\Theta,\Delta-\Theta)\right] \tag{7.19}$$

$$= \Delta^T \mathbf{H} \Theta - \left(F_{\mathbf{u}}^T \mathbf{u}\right) \Delta_{\mathbf{u}}^T \Theta_{\mathbf{u}} - \left(F_{\mathbf{v}}^T \mathbf{v}\right) \Delta_{\mathbf{v}}^T \Theta_{\mathbf{v}}. \tag{7.20}$$

If conjugate gradient (CG) is to be employed, we need the gradient on the Riemannian manifold $\mathcal{M}$, where the Riemannian metric

$$g_{\mathcal{M}}(\Delta,\Theta) = \Delta^T \Theta, \quad \Delta,\Theta \in \mathcal{T}\mathcal{M}$$

happens to be the Euclidean metric. This is because $\mathbf{u}$ and $\mathbf{v}$ are on spheres (specific case of Grassmann manifold). The gradient of F on $\mathcal{M}$ is simply

$$\nabla F = \left[\begin{array}{cccc} F_{\mathbf{u}} - \mathbf{u}\mathbf{u}^T F_{\mathbf{u}}, & F_{\mathbf{v}} - \mathbf{v}\mathbf{v}^T F_{\mathbf{v}}, & F_{\mathbf{w}}, & F_b \end{array}\right].$$

The (Euclidean) gradient of F_0 is

$$\nabla_e F_0 = \left[\begin{array}{cccc} \mathbf{g}_{\mathbf{u}}, & \mathbf{g}_{\mathbf{v}}, & \mathbf{g}_{\mathbf{w}}, & g_b \end{array}\right], \tag{7.21}$$

where by defining the residual vector

$$\mathbf{r} = \mathbf{T} + b - \mathbf{I} \tag{7.22}$$

and a tensor

$$\mathcal{G} = \mathcal{B} \times_{pixel} \mathbf{r}, \tag{7.23}$$

we have

$$\mathbf{g_u} = \left[\mathcal{G} \times_{ident} \mathbf{w}^T \times_{view} \mathbf{v}^T\right]_{(illum)} \tag{7.24}$$

$$\mathbf{g_v} = \left[\mathcal{G} \times_{illum} \mathbf{u}^T \times_{ident} \mathbf{w}^T\right]_{(view)} \tag{7.25}$$

$$\mathbf{g_w} = \left[\mathcal{G} \times_{view} \mathbf{v}^T \times_{illum} \mathbf{u}^T\right]_{(ident)}, \tag{7.26}$$

and $g_b = \mathbf{1}^T\mathbf{r}$ is the sum of image error (residual), where $\mathbf{1} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}$.

The regularization term is easy to differentiate, and the Euclidean gradient can be computed as

$$\nabla_e F = \nabla_e F_0 + \rho \begin{bmatrix} \mathbf{0}_U & \nabla_e R & \mathbf{0}_W + 1 \end{bmatrix} \tag{7.27}$$

To write the Hessian matrix, let us define two sets of matrices:

$$\mathbf{B_u} = \left[\mathcal{B} \times_{view} \mathbf{v}^T \times_{ident} \mathbf{w}^T\right]_{(pixel)} \tag{7.28}$$

$$\mathbf{B_v} = \left[\mathcal{B} \times_{ident} \mathbf{w}^T \times_{illum} \mathbf{u}^T\right]_{(pixel)} \tag{7.29}$$

$$\mathbf{B_w} = \left[\mathcal{B} \times_{illum} \mathbf{u}^T \times_{view} \mathbf{v}^T\right]_{(pixel)} \tag{7.30}$$

and

$$\mathbf{G_u} = \left[\mathcal{G} \times_{illum} \mathbf{u}^T\right]_{(view)} \tag{7.31}$$

$$\mathbf{G_v} = \left[\mathcal{G} \times_{view} \mathbf{v}^T\right]_{(ident)} \tag{7.32}$$

$$\mathbf{G_w} = \left[\mathcal{G} \times_{ident} \mathbf{w}^T\right]_{(illum)}. \tag{7.33}$$

Notice that $\mathbf{g_u} = \mathbf{B_u}^T\mathbf{r} = \mathbf{G_v}^T\mathbf{w}$, and similar relationships exist for $\mathbf{g_v}$ and $\mathbf{g_w}$ as well.

It can be shown that

$$\mathbf{H}_0 = \begin{bmatrix} \mathbf{B_u} & \mathbf{B_v} & \mathbf{B_w} & \mathbf{1} \end{bmatrix}^T \begin{bmatrix} \mathbf{B_u} & \mathbf{B_v} & \mathbf{B_w} & \mathbf{1} \end{bmatrix} + \begin{bmatrix} \mathbf{O} & \mathbf{G_w} & \mathbf{G_v}^T & 0 \\ \mathbf{G_w}^T & \mathbf{O} & \mathbf{G_u} & 0 \\ \mathbf{G_v} & \mathbf{G_u}^T & \mathbf{O} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (7.34)$$

The overall Hessian matrix is simply

$$\mathbf{H} = \mathbf{H}_0 + \rho \mathbf{H}_R, \quad (7.35)$$

where $\mathbf{H}_R$ is the Hessian matrix of (7.9) which has a simple diagonal form (that is why we employ$\|\mathbf{v}\|_p^p$ instead of $\|\mathbf{v}\|_p$). Note that the second term of (7.34) may be omitted because $\|\mathbf{r}\|$ is usually small, which would lend the Newton method we are going to use some flavor of Gauss-Newton.

7.3.1 Newton optimization on a Riemannian manifold

Similarly to the Euclidean case, the Newton step Δ is obtained as the tangent vector such that

$$\text{Hess } F(\Delta, \Theta) = -dF(\Theta) \quad (7.36)$$

for any $\Theta \in \mathcal{TM}$.

It is sufficient to find Δ satisfying (7.36) for $\{\mathbf{e}_i\}$ $(i = 1, 2, \ldots, U + V + W + 1)$, the bases of $\mathcal{TM}$. From (7.15) and (7.19), we can write (7.36) in matrix form:

$$\Theta^T \tilde{\mathbf{H}} \Delta = -\Theta^T \nabla_e F, \quad (7.37)$$

where

$$\tilde{\mathbf{H}} = \mathbf{H} - \text{diag}\left[\left(F_{\mathbf{u}}^T \mathbf{u}\right) \mathbf{1}_U, \quad \left(F_{\mathbf{v}}^T \mathbf{v}\right) \mathbf{1}_V, \quad \mathbf{0}_W, \quad 0 \right] \tag{7.38}$$

If we define a matrix $\mathbf{E}$ with column vectors being $\{\mathbf{e}_i\}$, and represent Δ as

$$\Delta = \mathbf{E} \cdot \mathbf{d}, \tag{7.39}$$

we obtain a linear equation from

$$\left(\mathbf{E}^T \tilde{\mathbf{H}} \mathbf{E}\right) \mathbf{d} = -\mathbf{E}^T \nabla_e F. \tag{7.40}$$

Solving (7.40) for $\mathbf{d}$, then we have the Newton step Δ immediately.

The basis matrix $\mathbf{E}$ can be easily constructed as $\mathbf{E} = \text{diag}\left(\mathbf{E}_{\mathbf{u}}, \mathbf{E}_{\mathbf{v}}, \mathbf{I}_W, 1\right)$. $\mathbf{E}_{\mathbf{u}}$ is a $U \times (U-1)$ matrix with columns being orthonormal tangent vectors of $\mathcal{S}^{U-1}$ at $\mathbf{u}$, which can be obtained from the Gram-Schmidt process as we know $\mathbf{E}_{\mathbf{u}}^T \mathbf{u} = \mathbf{0}$. Similarly we define $\mathbf{E}_{\mathbf{v}}$. Notice that now it is clear why we did not choose to constrain the magnitude of $\mathbf{w}$: our choice results in natural bases $\mathbf{I}_W$, reducing the computation in (7.40).

7.4 Concurrent Face Alignment and View/Illumination Estimation

Till now, we have assumed that the location of the input face is known; hence the face is normalized and cropped out so that the fitting algorithm discussed above can be directly applied to it. In this section, we discuss how to extend the approach to concurrently perform analysis and alignment.

Under the optimization framework for tensor-based face analysis, concurrent face alignment analysis is not much different from face analysis alone, as presented in the

previous section. To integrate alignment with the face analysis procedure, we just need to include the location parameters $\mathbf{p}$ into our optimization framework. Formally, we solve:

$$(\mathbf{u}^*, \mathbf{v}^*, \mathbf{w}^*, b^*, \mathbf{p}^*) = \underset{\mathbf{u},\mathbf{v},\mathbf{w},b,\mathbf{p}}{\arg\min} F_A(\mathbf{u}, \mathbf{v}, \mathbf{w}, b, \mathbf{p}; \rho), \text{ s.t. } \|\mathbf{u}\| = \|\mathbf{v}\| = 1, \tag{7.41}$$

where

$$F_A(\mathbf{u}, \mathbf{v}, \mathbf{w}, b, \mathbf{p}; \rho) = \tfrac{1}{2} \|\mathbf{T}(\mathbf{u}, \mathbf{v}, \mathbf{w}) + b - \mathbf{I}(\mathbf{p})\|^2 + \rho R(\mathbf{v}). \tag{7.42}$$

The gradient and Hessian of the new cost function with respect to all parameters other than $\mathbf{p}$ are the same as before, except that $\mathbf{I}(\mathbf{p})$ is now the face image extracted from the input image with location parameters $\mathbf{p}$, instead of a fixed image $\mathbf{I}$. Similar to face alignment with a linear appearance model, the gradient of F_A with respect to $\mathbf{p}$ can be written as

$$\mathbf{g}_\mathbf{p}^T = \left(\frac{\partial F_A}{\partial \mathbf{p}}\right)^T = -\left[\mathbf{T} + b - \mathbf{I}\right]^T \frac{\partial \mathbf{I}}{\partial \mathbf{p}}, \tag{7.43}$$

where $\frac{\partial \mathbf{I}}{\partial \mathbf{p}} = \begin{bmatrix} \frac{\partial \mathbf{I}_1}{\partial \mathbf{p}} & \cdots & \frac{\partial \mathbf{I}_N}{\partial \mathbf{p}} \end{bmatrix}^T$, the Jacobian of the warped and vectorized image $\mathbf{I}$, can be computed using the same technique introduced in Section 5.2.

To write the augmented function's Hessian, we define matrix

$$\mathbf{B}_\mathbf{p} = -\frac{\partial \mathbf{I}}{\partial \mathbf{p}} \tag{7.44}$$

Note that $\mathbf{g}_\mathbf{p} = \mathbf{B}_\mathbf{p}^T \mathbf{r}$, a similar expression to those for other parameters. Now the new Hessian matrix is

$$\mathbf{H}_A = \begin{bmatrix} \mathbf{B_u} & \mathbf{B_v} & \mathbf{B_w} & \mathbf{1} & \mathbf{B_p} \end{bmatrix}^T \begin{bmatrix} \mathbf{B_u} & \mathbf{B_v} & \mathbf{B_w} & \mathbf{1} & \mathbf{B_p} \end{bmatrix} \tag{7.45}$$

Note that, this way, we actually employed a Riemannian version of the

Gauss-Newton method.

7.5 Experiments

We conducted experiments on the "illum" subset of the CMU-PIE database [16]. We chose 5 views covering panning angle from $-90°$ to $90°$, which are representative views for the most common scenarios in face processing. We also chose 18 illuminations from the total of 21; this was due to some missing images in the released database. There are a total of 68 people in the database; we took the 9 people whose pictures are allowed to appear in publications according to PIE's user agreement. All images of the remaining 59 people were used to learn tensor appearance models. In the learned models, we keep the first 3 components for the illumination mode, which are sufficient to model face images lit by a single light source [70].

To make the comparison fair, the RTF algorithm was applied without sparsity regularization. Initialization is done by setting the view to frontal and illumination to the first basis (roughly corresponding to ambient illumination).

To quantitatively compare our RTF approach to the Rank-1 decomposition approach, we employed three criteria. First is the fitting cost, i.e. the difference between the input image and fitted image (measured in RMSE). The other two are normalized correlation of the estimated view and illumination coefficients respectively. The ground truth illumination coefficients are available in the mode matrix resulting from the learning procedure of the tensor model. Because the view mode was kept untransformed, the ground truth view coefficients will be all zeros except the one corresponding to the true view; therefore the normalized correlation between the estimated view coefficients and the ground truth is simply the estimated coefficient corresponding to the true view.

The mean and standard deviation values of the three criteria are summarized in Table 7.1. RTF is not only much more accurate but more robust as well.

Table 7.1: Performance on tensor face fitting of RTF and Rank-1 approach (5 dimensions in person mode).

Algorithm	**Fitting Cost**	**Illumination Score**	**View Score**
RTF	0.0729 ± 0.0239	0.9557 ± 0.1303	0.9587 ± 0.1100
Rank-1	0.2029 ± 0.1043	0.6370 ± 0.2964	0.7416 ± 0.2826

As we discussed, Rank-1 decomposition approach becomes less robust when the number of model parameters increases. In the second set of experiments, a tensor model consisting of 10 components in the person (identity) mode was used, while the other two modes were kept as before, hence doubling the free parameters Rank-1 decomposition approach has to solve in the LS step. The results are shown in Table 7.2.

Table 7.2: Performance on tensor face fitting of RTF and Rank-1 approach (10 dimensions in person mode).

Algorithm	**Fitting Cost**	**Illumination Score**	**View Score**
RTF	0.0696 ± 0.0230	0.9640 ± 0.1077	0.9652 ± 0.1019
Rank-1	0.2277 ± 0.1016	0.5678 ± 0.2804	0.7886 ± 0.2351

It is clear that the performance of the Rank-1 decomposition approach decreases when model complexity doubles. Although the accuracy of view estimation was better than in the previous case, the improvement was canceled out by the inferior illumination estimation, leading to higher average fitting cost. On the other hand, the proposed RTF approach remains as stable as before and even has some improvement. One reason for this stability is the fewer parameters the approach has to solve (i.e., $3 + 5 + 10 = 18$, which is only 5 more than in the previous setting). The improved accuracy is possibly due to the fact that increased dimension in person mode made the model more "expressive" and hence more accurate in fitting unseen faces.

To qualitatively display the results, 10 examples of fitting results obtained by RTF and Rank-1 approach are shown in Figure 7.3. These examples covered drastically varying views and illumination conditions. In each of these examples, the top row shows the input face image to fit and the initialization for RTF (which is always in frontal view and under ambient illumination). The second row displays the model fit from RTF and its de-lit (illumination-normalized) version. The corresponding images obtained from the Rank-1 approach are displayed in the bottom row.

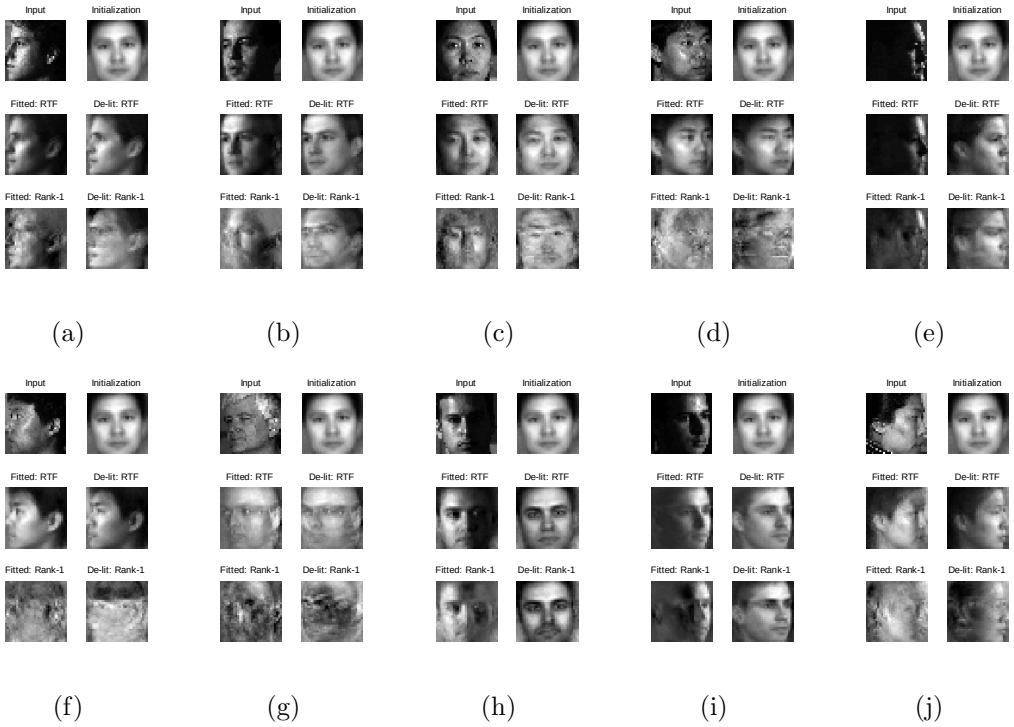


Figure 7.3: Examples of tensor appearance model fitting with RTF and Rank-1.

These results intuitively demonstrate RTF's capability of accurately estimating both illumination and view, even when the initial estimate departs far from the ground truth. On the contrary, the Rank-1 approach frequently fails to obtain a reasonable estimate although its linear step *theoretically* has a unique solution and does not need any initialization.

In the above comparison, for fairness with the Rank-1 approach which lacks the capability of employing regularization, we intentionally set the regularization term

of RTF to 0. Now we inspect the effect of the sparsity regularization. We changed the regularization weight ρ from 10^{-2} to 10^{3}, and recorded respective accuracy for the view and illumination coefficients, which are displayed in Figure 7.4. For a suitable range of ρ (about 1 to 10), the fitting accuracy is consistently improved compared to the case where no sparsity regularization is applied. Note that although the regularization is applied only to $\mathbf{v}$, the illumination coefficients $\mathbf{u}$ receive observable benefit from the regularization as well. Such results demonstrate the benefit of exploiting the anisotropic tensor model's sparse structure. It is also apparent that when ρ is too large, the performance breaks down. This is a direct result of over-regularizing where the main cost function is overwhelmed by the regularization term and cannot be effectively minimized.

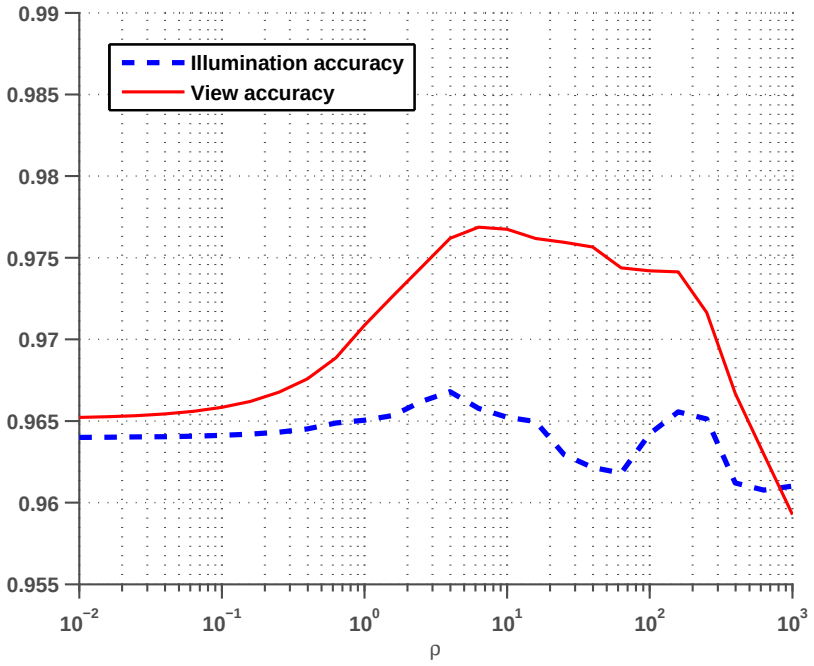


Figure 7.4: Effect of sparsity regularization in RTF.

Finally, we demonstrate the capability of RTF for concurrent face alignment and view/illumination estimation. Figure 7.5 shows a typical example. The tensor appearance model is initialized as before, except for the view mode where we set the

coefficients to be equal, rendering a uniformly mixed initial view. We found that in practice this strategy is more robust for fitting faces of different views than initializing with a certain fixed view. In Figure 7.5, from top to bottom are the initialization for the model and face location, the final state after RTF, and the convergence curve of fitting cost. The initial location (pose) was provided by the OpenCV face detector. Note that we intentionally rotated the input face image so that the initial pose (always upright) departs far enough from the ground truth to challenge the alignment algorithm. It can be observed that, although at the beginning the model has no knowledge about the view (it assumes that all views are equally possible) and illumination (assuming ambient illumination), through tens of iterations it gradually discovers the best estimate with the appearance model, and at the same time it normalizes the face's pose as well.

7.6 Summary

Our approach performs multi-view face alignment and meanwhile estimates the face's 3D view and illumination condition. The knowledge about view and illumination is useful for many applications, such as HCI. Good alignment is known to be extremely helpful for biometrics applications, both soft (i.e., estimating a person's gender, age, and ethnicity) and hard (face identification). In fact, under the tensor representation framework, our approach also outputs a view- and illumination-independent parameterization of the person's identity, i.e., $\mathbf{w}$, which can be used for face identification. Some existing works [62, 13] performed face recognition with similar identity representation, but classification was done by nearest neighbor using simple similarity metrics such as cosine; thus applying more powerful recognition algorithms developed in recent years, including the approaches proposed in this dissertation, is worth studying. While this falls out of the scope of

Figure 7.5: TensorAlign: Concurrent face alignment and view/illumination estimation with RTF.

this chapter, it is very interesting to investigate along this direction as one of the possibilities for future work.

Our approach is purely appearance-based and 2D in nature; however, it is able to understand factors (i.e., illumination and view) involving 3D information. Again, the prominent advantage our approach has over feature-based approaches belonging to the "flexible model" family (e.g. ASM, AAM and MM) is that our approach does not rely on precise localization of a collection of feature points — a task often, if not always, difficult for images of low resolution or low quality, which are frequently encountered in practical applications. On the other hand, the proposed approach has fewer parameters to optimize, resulting in benefits in both computation and robustness.

Certainly, our approach does not build a fine correspondence from model feature points to the input image, which may be important for certain tasks such as 3D view warping. However, the estimation of faces' location, viewpoint, illumination and identity constitute the most common face analysis tasks in practice, and for these the proposed approach is a promising solution.

Since ATM consists of submodels corresponding to discrete views, it might be considered similar to a model-based approach. However, in the fitting stage, the RTF method doe not need to traverse each of the views. Instead, the optimization procedure smartly leads the model towards the optimal view, from a probably very poor initialization. The regularization technique employed further improves the robustness in determining the optimal view.

The regularization technique is quite general. In this work the only regularization is the sparsity of the view coefficients; other priors on the tensor coefficients are not taken into consideration. The prior of the coefficients may be deduced from domain knowledge or learned from training data, and then integrated into our optimization framework as additional regularization terms. This would lead to more robust

coefficient estimation in the fitting stage, and is another direction deserving future research.

CHAPTER 8

CONCLUSIONS

Through this dissertation, we have studied key problems in face analysis, a practically important subarea of computer vision and recognition. Following the DAR (detection-alignment-recognition) paradigm, we have shown that appearance-based approaches are powerful tools toward the goal of building fully automatic face processing systems. We have especially discussed face recognition and alignment, introducing a series of effective algorithms. In this final chapter we briefly summarize our contributions and point out some directions for future research.

8.1 Contributions

The major contributions presented in this dissertation can be summarized into two categories: learning and modeling.

8.1.1 Learning: Boosting methods for visual recognition

Machine learning techniques have turned out to be effective tools for many highly complicated real-world problems, where it is hard or even impossible for humans to extract abstract knowledge and derive explicit rules to solve them. Asking computer algorithms to discover information from abundant data provides a chance to bypass this difficulty, so this is where learning techniques become valuable. Among recent advances in machine learning, boosting is undoubtably one of the most notable

developments. Motivated by earlier successful applications of boosting methods in computer vision, in this dissertation we dedicated several chapters discussing how to design boosting classifiers that are effective for facial recognition tasks, where the data lie in very high-dimensional spaces. Starting from MRC-Boosting which was developed for recognizing faces involving large intrapersonal variability, we showed that combining 1D classifiers in discriminatively learned subspaces under the boosting framework can lead to very effective classifiers. An important point made in that chapter is that direct learning may reach more discriminative features than those obtained by searching a pre-defined pool. As an extension, SODA-Boosting further consolidated this paradigm, demonstrating the power of second-order statistics in discovering features good for classification. Finally, BooMDA was proposed, refining ideas from earlier chapters, unifying the procedure of seeking optimal features based on second-order statistics, and introducing direct optimization techniques for 1D classifier learning.

8.1.2 Modeling: Appearance-based face alignment and analysis

Alignment is a significant step in most practical face processing systems. We showed the effectiveness of appearance-based approaches for this problem by proposing alignment algorithms of practical values. Compared to feature-based alignment algorithms, appearance-based approaches have the advantage in speed and robustness, especially when the input images are of low resolution or quality. As an example where appearance-based alignment can be practically very helpful, we provided an ensemble alignment algorithm useful for face database preprocessing. The algorithm is able to simultaneously align the whole ensemble of face images. This to some extent eliminates the costly and highly tedious procedure of manually labeling and aligning the whole database, which is necessary for most, if

not all, feature-based face alignment methods. Finally, in Chapter 7 we further demonstrated the power of appearance-based face alignment by proposing a purely appearance-based method that can handle drastically varying illuminations and viewpoints, which is usually thought to be possible only with a 3D model-based approach (which is necessarily feature-based). In developing these algorithms, techniques rooted in applied mathematics have turned out to be of great value. For instance, optimization methods — classic ones and more recent development such as geometric (Riemannian) optimization — have found their places when we devise effective face alignment algorithms — in fact, similar techniques helped our learning-based recognition algorithms as well, demonstrating their general applicability. As another example, in the context of multi-view face analysis, multi-linear algebra, which has received increasing attention in many engineering areas, was also shown to be a valuable tool.

8.2 Future Research

As stated in previous chapters, multi-view face analysis is becoming more and more important in many emerging applications, such as unconstrained face recognition and intelligent surveillance. In this dissertation we have presented an overview on general strategies, as well as introduced a novel alignment algorithm. In principle these results lay down a basis for multi-view analysis and build up a bridge between multi-view face processing and existing work on appearance-based frontal face analysis. However, to devise a practical, fully automatic system that is able to robustly analyze face images formed under highly variable conditions, there is still much work to be completed. On the other hand, although the multi-view analysis algorithm we presented is capable of interpreting most common and significant factors such as illumination and viewpoint, there are other facial properties that

may have theoretical or practical importance, e.g., aging and expression. It will be quite interesting if more work is to be done along this line.

The boosting family classification algorithms developed in this dissertation — although proposed and evaluated only in the scenario of face image analysis — are, as we pointed out, in nature quite general classification techniques, especially SODA-Boosting and BooMDA. Not relying on domain knowledge and predefined feature sets, they are expected to find more applications in areas other than computer vision. Even in the narrower scenario of face analysis there are, of course, many applications that we have not touched. For instance, we took gender recognition as an example of soft biometrics. Extending and applying the algorithms to solve other soft-biometrics problems, such as ethnicity group classification and age estimation, are also interesting in practice. On the other hand, since these classification algorithms are general, their application is not restricted to the pixel domain; in principle, they may work in any feature domain. Combining these algorithms with specific visual features, which have been shown effective for certain problems, provides another possibility to achieve higher recognition performance.

REFERENCES

[1] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[2] "Open Computer Vision Library." [Online]. Available: http://sourceforge.net/projects/opencvlibrary/

[3] Pittsburgh Pattern Recognition, "PittPatt Face Tracking SDK." [Online]. Available: http://www.pittpatt.com/products/

[4] T. F. Cootes and C. J. Taylor, "Statistical models of appearance for computer vision," University of Manchester, Wolfson Image Analysis Unit, Imaging Science and Biomedical Engineering, Manchester M13 9PT, United Kingdom, Tech. Rep., September 1999.

[5] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[6] M. J. Jones and T. Poggio, "Multidimensional morphable models: A framework for representing and matching object classes," *International Journal of Computer Vision*, vol. 29, no. 2, pp. 107–131, 1998.

[7] G. D. Hager and P. N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1025–1039, 1998.

[8] L. Sirovich and M. Kirby, "Low dimensional procedure for the characterization of human faces," *Journal of the Optical Society of America A*, vol. 4, no. 3, pp. 519–524, 1987.

[9] M. Turk and A. Pentland, "Eigenfaces for recognition," *Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–96, 1991.

[10] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[11] B. Moghaddam, T. Jebara, and A. Pentland, "Bayesian face recognition," *Pattern Recognition*, vol. 33, no. 11, pp. 1771–1782, 2000.

[12] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, 1997.

[13] V. Blanz and T. Vetter, "Face recognition based on fitting a 3d morphable model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1063–1074, 2003.

[14] X. Xu and T. S. Huang, "Face recognition with MRC-Boosting," in *10th IEEE International Conference on Computer Vision (ICCV 2005)*, vol. 2, 2005, pp. 1770–1777.

[15] X. Xu, Y. Rui, and T. S. Huang, "Recognizing faces in recorded meetings via MRC-Boosting," in *2006 IEEE International Conference on Multimedia & Expo (ICME 2006)*, 2006, pp. 1633–1636.

[16] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1615–1618, 2003.

[17] Z. Li and X. Tang, "Bayesian face recognition using support vector machine and face clustering," in *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, vol. 2. Los Alamitos, CA, USA: IEEE Computer Society, June 27 – July 2 2004, pp. 374–380.

[18] M. J. Jones and P. Viola, "Face recognition using boosted local features," Mitsubishi Electric Research Laboratories, Tech. Rep. TR2003-025, 2003.

[19] L. Zhang, S. Z. Li, Z. Y. Qu, and X. Huang, "Boosting local feature based classifiers for face recognition," in *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*, vol. 5. Washington, DC, USA: IEEE Computer Society, 2004, p. 87.

[20] M. Elad, Y. Hel-Or, and R. Keshet, "Pattern detection using a maximal rejection classifier," *Pattern Recognition Letters*, vol. 23, no. 12, pp. 1459–1471, 2002.

[21] C. Liu and H.-Y. Shum, "Kullback-Leibler boosting," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, vol. 1. Madison, Wisconsin: IEEE Computer Society, June 24–26 2003, p. 587.

[22] J. Tu, Z. Zhang, Z. Zeng, and T. Huang, "Face localization via hierarchical CONDENSATION with Fisher boosting feature selection," in *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*

(CVPR 2004), vol. 2. Los Alamitos, CA, USA: IEEE Computer Society, June 27 – July 2 2004, pp. 719–724.

[23] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," in *Proc. 14th International Conference on Machine Learning*. Morgan Kaufmann, 1997. [Online]. Available: http://citeseer.ist.psu.edu/153360.html pp. 322–330.

[24] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

[25] R. Cutler, Y. Rui, A. Gupta, J. Cadiz, I. Tashev, L.-w. He, A. Colburn, Z. Zhang, Z. Liu, and S. Silverberg, "Distributed meetings: A meeting capture and broadcasting system," in *MULTIMEDIA '02: Proceedings of the Tenth ACM International Conference on Multimedia*. New York, NY, USA: ACM, 2002, pp. 503–512.

[26] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.

[27] S. Romdhani, V. Blanz, and T. Vetter, "Face identification by fitting a 3D morphable model using linear shape and texture error functions," in *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*. London, UK: Springer-Verlag, 2002, pp. 3–19.

[28] X. Xu and T. S. Huang, "SODA-Boosting and its application to gender recognition," in *LNCS 4778: 2007 IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG), in conjunction with ICCV*. Rio de Janeiro, Brazil: Springer-Verlag, October 2007, pp. 193–204.

[29] B. A. Golomb, D. T. Lawrence, and T. J. Sejnowski, "Sexnet: A neural network identifies sex from human faces," in *NIPS-3: Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems 3*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990, pp. 572–577.

[30] G. W. Cottrell and J. Metcalfe, "EMPATH: face, emotion, and gender recognition using holons," in *NIPS-3: Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems 3*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990, pp. 564–571.

[31] S. Tamura, H. Kawai, and H. Mitsumoto, "Male/female identification from 86 very low resolution face images by neural network," *Pattern Recognition*, vol. 29, no. 2, pp. 331–335, 1996.

[32] S. Gutta, J. R. J. Huang, P. J. Phillips, and H. Wechsler, "Mixture of experts for classification of gender, ethnic origin, and pose of human faces," *IEEE Transaction on Neural Networks*, vol. 11, pp. 948–960, 2000.

[33] B. Moghaddam and M. Yang, "Learning gender with support faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 707–711, 2002.

[34] A. B. A. Graf and F. A. Wichmann, "Gender classification of human faces," in *BMCV '02: Proceedings of the Second International Workshop on Biologically Motivated Computer Vision*. London, UK: Springer-Verlag, 2002, pp. 491–500.

[35] G. Shakhnarovich, P. A. Viola, and B. Moghaddam, "A unified learning framework for real time face detection and classification," in *FGR '02: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*. Washington, DC, USA: IEEE Computer Society, 2002, p. 16.

[36] B. Wu, H. Ai, and C. Huang, "LUT-based AdaBoost for gender classification," in *4th International Conference on Audio and Video–Based Biometric Person Authentication (AVBPA 2003)*, 2003, pp. 104–110.

[37] S. Baluja and H. Rowley, "Boosting sex identification performance," in *IAAI'05: Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence*. AAAI Press, 2005, pp. 1508–1513.

[38] P. Viola and M. J. Jones, "Robust real-time object detection," in *IEEE Workshop on Statistical and Computational Theories of Vision*, July 13 2001.

[39] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd ed.)*. Wiley-Interscience, November 2000.

[40] T. Joachims, "Making large-scale support vector machine learning practical," in *Advances in Kernel Methods: Support Vector Learning*, B. Schlkopf, C. Burges, and A. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999, pp. 169–184.

[41] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York, NY: Springer, August 1999.

[42] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, December 1999.

[43] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *The Annals of Statistics*, vol. 28, no. 2, pp. 337–374, 2000. [Online]. Available: http://www.jstor.org/stable/2674028

[44] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. New York, NY: Springer, August 2006.

[45] P. Hallinan, "A low-dimensional representation of human faces for arbitrary lighting conditions," in *1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 1994)*. IEEE Computer Society, 1994, pp. 995–999.

[46] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218–233, 2003.

[47] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[48] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *ECCV '92: Proceedings of the Second European Conference on Computer Vision*. London, UK: Springer-Verlag, 1992, pp. 237–252.

[49] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *8th European Conference on Computer Vision*. Springer, 2004, pp. 25–36.

[50] W. Gao, B. Cao, S. Shan, X. Chen, D. Zhou, X. Zhang, and D. Zhao, "The CAS-PEAL large-scale chinese face database and baseline evaluations," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 38, pp. 149–161, 2008.

[51] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *SIGGRAPH '99: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194.

[52] S. Romdhani and T. Vetter, "Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior," in *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 986–993.

[53] J. Xiao, S. Baker, I. Matthews, and T. Kanade, "Real-time combined 2d+3d active appearance models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 535–542.

[54] Y. Hu, Z. Zhang, X. Xu, Y. Fu, and T. S. Huang, "Building large scale 3d face database for face analysis," in *MCAM 2007: International Workshop on Multimedia Content Analysis and Mining*, Weihai, China, June 2007, pp. 343–350.

[55] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[56] A. Kapteyn, H. Neudecker, and T. Wansbeek, "An approach to n-mode components analysis," *Psychometrika*, vol. 51, no. 2, pp. 269–275, 1986.

[57] L. de Lathauwer, B. de Moor, and J. Vandewalle, "On the best rank-1 and rank-$(\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_N)$ approximation of higher-order tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[58] L. de Lathauwer, B. de Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[59] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I.* London, UK: Springer-Verlag, 2002, pp. 447–460.

[60] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear image analysis for facial recognition," in *Proceedings of International Conference on Pattern Recognition (ICPR 2002)*, vol. 2, Quebec City, Canada, August 2002. [Online]. Available: http://www.mrl.nyu.edu/~maov/tensorfaces/icpr02.pdf pp. 511–514.

[61] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear subspace analysis of image ensembles," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, vol. 2. Madison, Wisconsin: IEEE Computer Society, June 24–26 2003, p. 93.

[62] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear independent components analysis," in *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1.* Washington, DC, USA: IEEE Computer Society, 2005, pp. 547–553.

[63] M. Vasilescu and D. Terzopoulos, "Multilinear projection for appearance-based recognition in the tensor framework," in *IEEE 11th International Conference on Computer Vision (ICCV 2007)*, 2007, pp. 1–8.

[64] P. N. Belhumeur and D. J. Kriegman, "What is the set of images of an object under all possible illumination conditions?" *International Journal of Computer Vision*, vol. 28, no. 3, pp. 245–260, 1998.

[65] D. L. Donoho, "For most large underdetermined systems of equations, the minimal l^1-norm near-solution approximates the sparsest near-solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 7, pp. 907–934, 2006.

[66] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[67] S. T. Smith, "Geometric optimization methods for adaptive filtering," Ph.D. dissertation, Harvard University, Cambridge, MA, USA, 1993.

[68] A. Edelman, T. As, A. Arias, Steven, and T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, pp. 303–353, 1998.

[69] Y. Ma, "A differential geometric approach to computer vision and its applications in control," Ph.D. dissertation, University of California, Berkeley, 2000.

[70] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

AUTHOR'S BIOGRAPHY

Xun Xu was born in Jiangsu, China. He received his bachelor and master degrees from the Department of Automation, Tsinghua University, Beijing, China, in 2000 and 2003 respectively. He then joined the Beckman Institute and Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign for his Ph.D. study, under the supervision of Professor Thomas S. Huang, and obtained the degree in 2010. Dr. Xu's research interests fall in computer vision, image/video processing, pattern recognition, and machine learning as well as their application to real-world problems such as biometrics, human-computer interaction, intelligent surveillance and medical imaging. In his doctoral research, he studied algorithms for appearance-based modeling and learning of the human face and their application to biometrics.